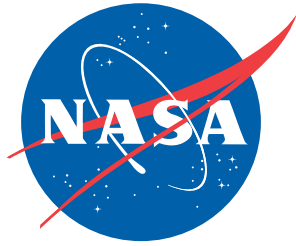


NASA/TM-2011-217063
NESC-RP-09-00530



Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis, Phase 2 Results

*Daniel G. Murri/NESC
Langley Research Center, Hampton, Virginia*

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

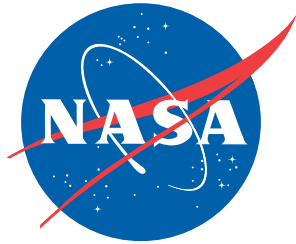
- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at 443-757-5803
- Phone the NASA STI Help Desk at 443-757-5802
- Write to:
NASA STI Help Desk
NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320

NASA/TM-2011-217063
NESC-RP-09-00530



Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis, Phase 2 Results

*Daniel G. Murri/NESC
Langley Research Center, Hampton, Virginia*

National Aeronautics and
Space Administration


Langley Research Center
Hampton, Virginia 23681-2199

February 2011

The use of trademarks or names of manufacturers in the report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.


Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 1 of 73

Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis, Phase 2 Results

February 3, 2011

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 2 of 73

Report Approval and Revision History

NOTE: This document was approved at the February 3, 2011, NRB. This document was submitted to the NESC Director on February 10, 2011, for configuration control.

Approved Version:	<i>Original Signature on File</i>	2/10/11
1.0	NESC Director	Date

Version	Description of Revision	Office of Primary Responsibility	Effective Date
1.0	Initial Release	Mr. Daniel G. Murri, NASA Technical Fellow for Flight Mechanics	2/3/2011


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 3 of 73

Table of Contents

Volume I: Technical Assessment Report

1.0	Notification and Authorization	5
2.0	Signature Page.....	6
3.0	Team List	7
4.0	Executive Summary	8
5.0	Models and Script Descriptions	9
5.1	Multi-mode EKF	10
5.1.1	POST2 Sensor Models for mEKF.....	11
5.1.1.1	IMU and Gyroscope Model	11
5.1.1.2	Star Tracker Model	12
5.1.1.3	Altimeter and Velocimeter Models.....	13
5.1.2	POST2 Inputs/Outputs for mEKF.....	13
5.2	Aerodynamic Uncertainty Model.....	17
5.3	Terminal Point Control Guidance Algorithm	22
5.4	Aerobraking Mission Design Module.....	27
5.5	Earth Environment Models	30
5.5.1	2010 Earth Global Reference Atmosphere Model	30
5.5.2	Earth Gravity Model 96	36
5.6	Attitude Control Models	36
5.6.1	Pseudo-Controller for 3-axis Attitude Control.....	37
5.6.2	Second Order Actuator Model	40
6.0	Observation and NESC Recommendation	40
7.0	Other Deliverables	41
8.0	Definition of Terms	41
9.0	Acronyms List	42
10.0	References.....	43

List of Tables

Table 5.0.1.	Responsible Individuals for Models Implemented and Report Sections	10
Table 5.1-1.	POST2 Inputs for the mEKF Algorithm	13
Table 5.1-2.	POST2 Outputs for the mEKF Algorithm	17
Table 5.2-1.	POST2 Inputs for the Aerodynamic Dispersion Model	19
Table 5.2-2.	POST2 Outputs from the Aerodynamic Dispersion Model	22
Table 5.3-1.	POST2 Inputs for the TPC Guidance Algorithm	25
Table 5.3-2.	POST2 Outputs for the TPC Guidance Algorithm	27
Table 5.4-1.	POST2 Inputs for the Aerobraking Mission Design Module	28
Table 5.4-2.	POST2 Table Inputs for the Aerobraking Mission Design Module	29
Table 5.4-3.	POST2 Outputs for the Aerobraking Mission Design Module	30
Table 5.5-1.	POST2 Inputs for the Earth-GRAM 2010 Model	31
Table 5.5-2.	POST2 Tables for the Earth-GRAM 2010 Model	34



	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 4 of 73

Table 5.5-3.	POST2 Outputs for the Earth-GRAM 2010 Model	35
Table 5.6.1-1.	POST2 Inputs for the 3-Axis Pseudo-Controller Model.....	38
Table 5.6.1-2.	POST2 Outputs for the 3-Axis Pseudo-Controller Model	39

Volume II: Appendices

Appendix A. Multi-Mode Extended Kalman Filter

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 5 of 73

Volume I: Technical Assessment Report


1.0 Notification and Authorization

Mr. Daniel Murri, NASA Technical Fellow for Flight Mechanics, requested that the NASA Engineering and Safety Center (NESC) support an inquiry from Mr. Harold Bell, Director Advanced Planning and Analysis Division, NASA Office of Chief Engineer, to establish the Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis assessment. The principal focus of the proposed activity was to develop a simulation framework and a set of validated and documented subsystem models and scripts to allow rapid evaluation of EDL characteristics in systems analysis studies, preliminary design, mission development and execution, and time-critical assessments.

The Initial Evaluation for this assessment was approved by the NESC Review Board (NRB) on March 12, 2009. Mr. Daniel Murri was selected to lead this assessment. The Assessment Plan was presented and approved by the NRB on March 26, 2009. The Phase 1 Stakeholder Outbrief and final report were approved by the NRB on November 5 and December 17, 2009, respectively.


The Phase 2 Assessment Plan was approved by the NRB on December 17, 2009. The Phase 2 final report was presented to the NRB for approval on February 3, 2011.

The key stakeholders for this activity are Mr. Harold Bell; Mr. Thomas Zang, EDL-Systems Analysis (EDL-SA) Team Lead; and the NESC.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 7 of 73

3.0 Team List

Name	Discipline	Organization
Core Team		
Dan Murri	NESC Team Lead	LaRC
Richard Powell	Team Co-Lead	LaRC/AMA
Scott Striepe	Simulation Lead	LaRC
Eric Queen	Controls Lead	LaRC
Mark Schoenenberger	Aerodynamics Lead	LaRC
Alicia Cianciolo	Environments Lead	LaRC
John Wagner	Mass Lead	LaRC/NIA
Loreyna Yeung	Scripts Lead	LaRC/ATK
John Aguirre	Software	LaRC/Vigyan
Carole Garrison	Software	LaRC/ATK
Michael Kelly	Backup Principal Engineer	LaRC
Dave Kinney	Aerodynamics	ARC
Jack Mulqueen	Aerospace Engineer	MSFC
Jill Prince	Flight Mechanics	LaRC
David Way	Flight Mechanics	LaRC
Carlos Westhelle	Guidance	JSC
Robert Bishop	Navigation	Marquette University
Consultants		
Phil Calhoun	Controls	GSFC
Cornelius Dennehy	NASA Technical Fellow for GN&C	GSFC
Michael Hagopian	Senior AETD Engineer	GSFC
Kurt Kloesel	Electrical Engineer	DFRC
Gary Mosier	Aerospace Engineer	GSFC
Martin Steele	Computer Engineer	KSC
Administrative Support		
Tricia Johnson	MTSO Program Analyst	LaRC
Melinda Meredith	Project Coordinator	LaRC/ATK
Linda Burgess	Planning and Control Analyst	LaRC/ATK
Erin Moran	Technical Writer	LaRC/ATK

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 8 of 73


4.0 Executive Summary

The NASA Engineering and Safety Center (NESC) was requested to establish the Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis assessment, which involved development of an enhanced simulation architecture using the Program to Optimize Simulated Trajectories II (POST2)¹ simulation tool. The assessment was requested to enhance the capability of the Agency to provide rapid evaluation of EDL characteristics in systems analysis studies, preliminary design, mission development and execution, and time-critical assessments. Many of the new simulation framework capabilities were developed to support the Agency EDL-Systems Analysis (SA) team that is conducting studies of the technologies and architectures that are required to enable human and higher mass robotic missions to Mars.

The EDL-SA studies have been conducted using POST2, but with a variety of ad-hoc and undocumented sub-system models (e.g., mass, aerodynamics, atmosphere, guidance, control) and scripts. During Phase 1 of the assessment [ref.1], the NESC team developed a simulation framework and a set of validated and documented sub-system models (including mass models, a pseudo bank angle controller, aerodynamic trim, aerodynamic data models, atmospheric models, guidance algorithms) and scripts. Phase 2 augments the previous effort with several models that were not included in Phase 1, which add to the simulation support of the EDL-SA team, and Agency and NESC rapid assessments. These models include a navigation filter for EDL, aerodynamic uncertainty, second order actuator model, aerocapture guidance algorithm, aerobraking mission design, Earth-Global Reference Atmosphere Model (GRAM) 2010 [ref. 2], Earth Gravity Model (EGM) 96, and a three-axis pseudo-controller. A set of input cases was generated for the models and is maintained with the simulation. This test suite will be used to confirm that future software models added to the simulation do not adversely affect the operation of the existing models added from this assessment. In the POST2 simulation environment, the simulation framework developed during this assessment is referred to as Rapid EDL Analysis Simulation (REDLAS).

The observations and NESC recommendations are discussed in Section 6.0. Overall, the main objective to increase the Agency's ability to rapidly evaluate EDL characteristics in systems analysis studies, preliminary design, mission development and execution, and time-critical assessments was accomplished. It is recommended that the custodians of the POST2 simulation framework, the Atmospheric Flight and Entry Systems Branch at Langley Research Center (LaRC), should retain and maintain the REDLAS models, test cases, and scripts for use by current and future programs.

¹ POST2 is an ITAR restricted code and may only be distributed to approved persons. <https://post2.larc.nasa.gov/>

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 9 of 73

5.0 Models and Script Descriptions

The NESC was requested to establish the Simulation Framework for Rapid EDL Analysis assessment, which involved development of an enhanced simulation architecture using the POST2 simulation tool. The simulation developed in Phase 1 included mass models, a pseudo controller, aerodynamic models, atmospheric models, guidance algorithms, and various support scripts. Phase 2 augments the previous effort with several models that were not included in Phase 1, which add to the simulation support of the EDL-SA team and Agency and NESC rapid assessments. A set of input cases was generated for the models and is included with the simulation. This test suite can be used to confirm that future software models added to the simulation do not adversely affect the operation of the existing models from this assessment.

The models described below were developed and/or implemented by the NESC team. This report is outlined and discussed as follows:

- Section 5.1: multimode Extended Kalman Filter (mEKF)
- Section 5.2: Aerodynamic Uncertainty Models
- Section 5.3: Terminal Point Control (TPC) Guidance Algorithm
- Section 5.4: Aerobraking Mission Design Module
- Section 5.5: Earth Environment Models (atmosphere and gravity)
- Section 5.6: Attitude Control Models (3-axis pseudo-controller (3APC), second order actuator model)

Table 5.0.1 lists the individuals responsible for the various models and sections of this report.


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 10 of 73


Table 5.0.1. Responsible Individuals for Models Implemented and Report Sections

Model Description	Section(s)	Responsible Individual
mEKF	5.1 and Appendix A	Robert Bishop
POST2 sensor models and input/output	5.1.1 – 5.1.2	Scott Striepe
Aerodynamics Uncertainty Models	5.2	Mark Schoenenberger
POST2 Aerodynamic Uncertainty Inputs & Outputs	5.2.1	Scott Striepe
TPC Guidance Algorithms	5.3	Eric Queen
Aerobraking Mission Design Module	5.4	Alicia Cianciolo
Earth-GRAM 2010	5.5.1	John Aguirre/Dick Powell
EGM96 POST2 inputs	5.5.2	Scott Striepe
Three-Axis Pseudo-Controller	5.6.1	Eric Queen/Scott Striepe
Second Order Actuator Model	5.6.2	Eric Queen

5.1 Multi-mode EKF

A significant area of investigation supporting the NESC’s Simulation Framework for Rapid EDL Analysis is the development of a mEKF for navigation. While using perfect navigation (i.e., perfect knowledge of the true state) with guidance systems is a reasonable first design step, errors from imperfect state knowledge help determine the robustness of the guidance and vehicle design. The mEKF navigation filter architecture for application to EDL is consistent with the methodology being used for flight systems that are currently under development.

A key element of the mEKF is sensor error modeling. The objective is to construct a generalized information structure to represent sensors that encompass the requisite data required to compute and process mEKF measurements. The mEKF navigation filter architecture can be used in applications of EDL using a standard set of measurements including altimeters, velocimeters, star trackers, and inertial measurement units. The navigation architecture main computational unit is the mEKF algorithm employing a dual-inertial state implementation with integrated

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 11 of 73

position, velocity, and attitude. The filter described here is readily re-configurable to accommodate differing mission scenarios (e.g., lunar descent or Mars entry).

Also, several sensor models were included in POST2 to provide mEKF measurements. These models create measurements from values calculated during the simulation and applying errors such as bias, scale factors, and random noise. The inertial measurement unit (IMU) model generates Δ -V and Δ - θ measurements sensed acceleration and angular velocity. The vehicle altitude and velocity relative to the surface are used for the altimeter and velocimeter, respectively. The star tracker model uses the inertial to body transformation to define attitude measurements.

Details of the theory and equations used for the mEKF and its sensor error models are provided in Appendix A. Sensor models in POST2 to provide measurement data are given in Section 5.1.1. Inputs and outputs associated with these models as implemented in POST2 are identified in Section 5.1.2.

5.1.1 POST2 Sensor Models for mEKF

5.1.1.1 IMU and Gyroscope Model


The POST2-developed IMU model is a statistically based accelerometer and gyroscope model. The accelerometer model takes the true body acceleration, with a random bias, noise, and scale factor errors to generate a measurement of accelerations that are then converted to Δ -V before being passed to mEKF for processing. The acceleration measurement model is given by:

$$\vec{a}_{acc} = (I_{3 \times 3} + SF_{acc})\vec{a}_{env} + b_{acc} + \eta_{acc}, \quad (5.1-1)$$

and

$$SF_{acc} = \begin{bmatrix} SF_{acc(x)} & 0 & 0 \\ 0 & SF_{acc(y)} & 0 \\ 0 & 0 & SF_{acc(z)} \end{bmatrix}, \quad \vec{b}_{acc} = \begin{bmatrix} b_{acc(x)} \\ b_{acc(y)} \\ b_{acc(z)} \end{bmatrix}, \quad \vec{\eta}_{acc} = \begin{bmatrix} \eta_{acc(x)} \\ \eta_{acc(y)} \\ \eta_{acc(z)} \end{bmatrix} \quad (5.1-2)$$

where \vec{a}_{acc} is the accelerometer measured acceleration vector in m/s^2 ; \vec{a}_{env} is the true body sensed acceleration vector in m/s^2 ; $I_{3 \times 3}$ is the identity matrix; SF_{acc} is the accelerometer scale factor diagonal matrix; \vec{b}_{acc} is the accelerometer bias; and $\vec{\eta}_{acc}$ is the random noise. The accelerometer measurement updates begin at the start of the trajectory, and continue until touchdown at a user input rate. The integration of the sensed acceleration across a measurement interval defines the Δ -V:

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 12 of 73

$$\Delta - \bar{V})_k = \int_{t_{k-1}}^{t_k} \bar{a}_{acc} \cdot dt \quad (5.1-3)$$

and the accumulated Δ -V is the sum of the Δ -V generated between updates to the navigation filter.

Similarly, the gyroscope model takes the true body angular rates, with a random noise, bias and scale factor errors to generate a measurement of angular rates that are converted to Δ - θ before being passed to navigation. Similar to the accelerometer, Equations (5.1-1), (5.1-2), and (5.1-3), the gyroscope angular rate measurement model is:

$$\bar{\omega}_{gyro} = (I_{3 \times 3} + SF_{gyro}) \bar{\omega}_{env} + \bar{b}_{gyro} + \bar{\eta}_{gyro} \quad (5.1-4)$$

where $\bar{\omega}_{gyro}$ is the gyroscope measured angular rate vector in rad/s; $\bar{\omega}_{env}$ is the true (environment) body angular rate vector in rad/s; SF_{gyro} is the gyroscope scale factor diagonal matrix; b_{gyro} is the gyroscope bias; and η_{gyro} is the random noise. The integrated angular velocity, $\bar{\omega}_{gyro}$, is integrated across a measurement interval to provide the Δ - θ measurement:

$$\Delta - \bar{\theta})_k = \int_{t_{k-1}}^{t_k} \bar{\omega}_{gyro} \cdot dt \quad (5.1-5)$$


which is then accumulated (summed) until the next call to the mEKF. The gyroscope measurement updates also begin at the start of the trajectory and continue until touchdown at a user-defined rate.

5.1.1.2 Star Tracker Model

The POST2-developed star tracker measurement model is a statistically based model. The star tracker model takes the true attitude quaternion, bias and noise to generate a star tracker measurement that is passed to the mEKF for processing as an external measurement. The noise ($\bar{\eta}_{st}$) and bias (\bar{b}_{st}) are used to calculate an error quaternion ($\bar{q}_{e(b,\eta)}$):

$$\bar{q}_{e(b,\eta)} = \begin{bmatrix} \cos\left(\frac{\beta_{st}}{2}\right) \\ \bar{\beta}_{st} \sin\left(\frac{\beta_{st}}{2}\right) \end{bmatrix}, \quad (5.1-6)$$

where

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 13 of 73

$$\begin{aligned}\vec{\beta}_{st} &= \vec{\eta}_{st} + \vec{b}_{st}, \\ \beta_{st} &= \|\vec{\beta}_{st}\|\end{aligned}\tag{5.1-7}$$

The star tracker measured attitude quaternion (\bar{q}_{st}) is determined as the product of the true quaternion (\bar{q}_{env}) and error quaternion ($\bar{q}_{e(b,\eta)}$) yielding

$$\bar{q}_{st} = \bar{q}_{e(b,\eta)} \bar{q}_{env} \quad \square \tag{5.1-8}$$

5.1.1.3 Altimeter and Velocimeter Models

☐ The measured altitude (h_{sensor}) determined by the altimeter sensor is modeled in POST2 as:

$$h_{sensor} = h_{truth} (1 + SF_h) + (p_{noise} h_{truth} + h_{noise}) + h_{bias} \tag{5.1-9}$$

where h_{truth} is the true altitude, and p_{noise} , h_{noise} , h_{bias} , and SF_h are the noise percentage, noise addition, bias, and scale factor specifications, respectively. The sensor altitude measurement updates begin at a user defined truth altitude above the planet surface at a data rate also input by the user. The sensor velocity is separated into planet relative velocity vector components expressed in the body coordinate system. The sensor velocity for each component (v_{sensor}) independently determined by the velocimeter is modeled as:

$$v_{sensor} = v_{truth} (1 + SF_v) + v_{noise} + v_{bias} \tag{5.1-10}$$


where v_{truth} is the true relative velocity component being modeled, and v_{noise} , v_{bias} , and SF_v are the noise, bias, and scale factor specifications, respectively.

5.1.2 POST2 Inputs/Outputs for mEKF


The inputs and outputs for use with the mEKF navigation filter have been integrated into the POST2 input/output dictionary. The inputs are identified in Table 5.1-1 and the outputs are identified in Table 5.1-2.

Table 5.1-1. POST2 Inputs for the mEKF Algorithm


Input Symbol	Units / Type	Stored Value	Definition
EKF_NX	integer	0	Number of states in the filter
EKF_NP	integer	0	Number of states in the covariance

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 14 of 73

Input Symbol	Units / Type	Stored Value	Definition
EKF_NM	integer	0	Number of measurements
EKF_NAV_RATE	Hz	0.0	Update rate of the mEKF
EKF_NAV_DT	s	0.0	Inverse of the navigation rate
EKF_Q_F_S_0, EKF_Q_F_S_1, EKF_Q_F_S_2, EKF_Q_F_S_3	nd	1.0, 0.0, 0.0, 0.0	Quaternion of orientation of planet-fixed reference frame with respect to the planet surface reference frame
EKF_R_REF_F_X, EKF_R_REF_F_Y, EKF_R_REF_F_Z	m	0.0, 0.0, 0.0	Location of the surface reference frame origin in the planet fixed-frame
EKF_OMEGA_FI_F_X, EKF_OMEGA_FI_F_Y, EKF_OMEGA_FI_F_Z	m	0.0, 0.0, 0.0	Spin axis of planet or moon at epoch
EKF_Q_IE_F_0, EKF_Q_IE_F_1, EKF_Q_IE_F_2, EKF_Q_IE_F_3	nd	1.0, 0.0, 0.0, 0.0	Quaternion defining orientation of the inertial reference frame with respect to the planet-fixed reference frame at epoch
EKF_QIMU_i_j, i=1,6 j=1,6	nd	0.0	IMU process noise matrix
EKF_QTUNE_i_i, i=1,22	nd	0.0	Tuning process noise matrix
EKF_TE	s	0.0	Epoch associated with Q_IE_F
EKF_MOON_AE	m	1737400	Radius of the moon for gravity calculations
EKF_MOON_APO	m	1735970	Polar radius of the moon
EKF_MOON_AEQ	m	1738140	Equatorial radius of the moon
EKF_MOON_J2	nd	203.32E-06	Gravity J2 term of the moon
EKF_MOON_MU	m ³ /s ²	4902.8E09	Gravitational constant of the moon
EKF_MOON_OMEGA	rad/s	2.6617E-06	Rotation rate of the moon
EKF_MARS_AE	m	3397200	Radius of Mars for gravity calculations
EKF_MARS_APO	m	3376200	Polar radius of Mars
EKF_MARS_AEQ	m	3396190	Equatorial radius of Mars
EKF_MARS_J2	nd	0.0020	Gravity J2 term of Mars
EKF_MARS_MU	m ³ /s ²	4.2828376383e+13	Gravitational constant of Mars
EKF_MARS_OMEGA	rad/s	7.088218e-05	Rotation rate of Mars
EKF_VEL_BIAS_X, EKF_VEL_BIAS_Y, EKF_VEL_BIAS_Z	m/s	0.0, 0.0, 0.0	Velocimeter measurement bias in the body X, Y, and Z axes

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 15 of 73

Input Symbol	Units / Type	Stored Value	Definition
EKF_VEL_NOISE_COV_X_X, EKF_VEL_NOISE_COV_X_Y, EKF_VEL_NOISE_COV_X_Z, EKF_VEL_NOISE_COV_Y_X, EKF_VEL_NOISE_COV_Y_Y, EKF_VEL_NOISE_COV_Y_Z, EKF_VEL_NOISE_COV_Z_X, EKF_VEL_NOISE_COV_Z_Y, EKF_VEL_NOISE_COV_Z_Z	nd	0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0	Velocimeter measurement noise covariance matrix
EKF_ATT_C_BIAS_X, EKF_ATT_C_BIAS_Y, EKF_ATT_C_BIAS_Z	m/s	0.0, 0.0, 0.0	Attitude camera measurement bias in the body X, Y, and Z axes
EKF_ATT_C_NOISE_COV_X_X, EKF_ATT_C_NOISE_COV_X_Y, EKF_ATT_C_NOISE_COV_X_Z, EKF_ATT_C_NOISE_COV_Y_X, EKF_ATT_C_NOISE_COV_Y_Y, EKF_ATT_C_NOISE_COV_Y_Z, EKF_ATT_C_NOISE_COV_Z_X, EKF_ATT_C_NOISE_COV_Z_Y, EKF_ATT_C_NOISE_COV_Z_Z	nd	0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0	Attitude camera measurement noise covariance matrix
EKF_ALT_BIAS	m	0.0	Spherical altimeter measurement bias

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 16 of 73

Input Symbol	Units / Type	Stored Value	Definition
EKF_ALT_NOISE_COV	nd	0.0	Spherical altimeter measurement noise covariance
EKF_ACCEL_SPEC_BIAS	m/s	0.0	IMU Δ -V bias standard deviation ($1-\sigma$)
EKF_ACCEL_SPEC_NOISE	nd	0.0	IMU Δ -V noise covariance
EKF_GYRO_SPEC_BIAS	rad	0.0	IMU Δ - θ bias standard deviation ($1-\sigma$)
EKF_GYRO_SPEC_NOISE	nd	0.0	IMU Δ - θ noise covariance
EKF_XI_DELTA_ERR, EKF_YI_DELTA_ERR, EKF_ZI_DELTA_ERR	m	0.0, 0.0, 0.0	Initial navigation position error input as distance from true inertial state
EKF_VXI_DELTA_ERR, EKF_VYI_DELTA_ERR, EKF_VZI_DELTA_ERR	m/s	0.0, 0.0, 0.0	Initial navigation velocity error input relative to true inertial state
EKF_ERR_Q_0, EKF_ERR_Q_1, EKF_ERR_Q_2, EKF_ERR_Q_0	nd	1.0, 0.0, 0.0, 0.0	Initial quaternion defining the inertial navigation attitude error


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 17 of 73


Table 5.1-2. POST2 Outputs for the mEKF Algorithm

Output Symbol	Type/ Units	Definition
ekf_x_km_x, ekf_x_km_y, ekf_x_km_z	m	Propagated navigation inertial position estimate
ekf_x_km_vx, ekf_x_km_vy, ekf_x_km_vz	m/s	Propagated navigation inertial velocity estimate
ekf_x_km_q0, ekf_x_km_q1, ekf_x_km_q2, ekf_x_km_q3	nd	Propagated navigation inertial to body quaternion estimate
ekf_P_km	m^2/s^2	Propagated navigation covariance matrix
ekf_x_kp_x, ekf_x_kp_y, ekf_x_kp_z	m	Navigation inertial position estimate after update
ekf_x_kp_vx, ekf_x_kp_vy, ekf_x_kp_vz	m/s	Navigation inertial velocity estimate after update
ekf_x_kp_q0, ekf_x_kp_q1, ekf_x_kp_q2, ekf_x_kp_q3	nd	Navigation inertial to body quaternion estimate after update
ekf_P_kp	m^2/s^2	Navigation covariance matrix after update

5.2 Aerodynamic Uncertainty Model

Part of the Phase 1 effort was to include several aerodynamic models (i.e., the 70-, 60-, and 45-degree sphere-cone aerodynamic database routines). However, the uncertainty values (used in Monte Carlo assessments) were required to be user input. This approach requires the user to have detailed knowledge of the uncertainty values and their application for the given database being used. To avoid user input errors associated with aerodynamic uncertainties, the method described below was developed wherein the uncertainty value and application are held within the aerodynamic database subroutine. The user inputs the desired standard deviation value (i.e., sigma value) of the uncertainty desired. The actual dispersion value and its application to the nominal aerodynamic coefficient values throughout the trajectory simulation are computed within the aerodynamic subroutine. This method is now used for the dispersion calculation associated with the Phase 1 aerodynamic models listed above.

The dispersions are input in specific flight regimes. The static coefficient dispersions are defined in three main regions of the flight regime: free molecular, hypersonic continuum, and supersonic continuum. The free molecular flight regime is defined by Knudsen number values ≥ 1000

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 18 of 73

(indicated by `_UNC1` in the variable name). For Knudsen number < 1000 , the hypersonic continuum is defined by Mach numbers between 5 and 10 (`_UNC2` in variable name) and the supersonic continuum region is defined by $\text{Mach} \leq 5$ (`_UNC3` in variable name). The transitional aerodynamic uncertainties are determined using linear interpolation. Interpolation using log base 10 of the Knudsen number is applied between free molecular and supersonic continuum dispersion values. Whereas, interpolation using Mach number is utilized between hypersonic and supersonic continuum dispersion values. The only difference for dynamic derivative coefficient dispersions is that the hypersonic continuum is defined by Mach numbers below 6, and supersonic continuum is Mach number 3 and less.

The dispersions are determined in the `DISPERSION` subroutine (in the file `dispersion.f` in the `redl_Aero` directory). This routine receives nominal aerodynamic coefficients (at respective database moment reference points) from the aerodynamic database routines and transforms moments to the center of gravity (`cg`). The aerodynamic coefficients are then dispersed about the `cg` using 3-sigma values [refs. 3, 4, 5] scaled by the user input (as described in Table 5.2-1). The dispersed aerodynamic coefficients are then transformed to the moment reference point for correct dispersion application by `POST2` computations, which include the contribution of the aerodynamic forces to the aerodynamic moments. User input values of zero apply no dispersion and values may range from -1 to 1. Three-sigma values are multiplied by these input values (i.e., an input of 1 applies a +3 sigma dispersion as indicated in Table 5.2-1).

The general variables in Table 5.2-1 are associated with the aerodynamic inputs, whereas Table 5.2-2 are the outputs in `POST2`. As shown parenthetically in Table 5.2-1, similar dispersed values are available for the other flight regimes with the variable name changed (specifically the `_UNC1` portion) as described previously. The only exception to this information is the `CMQ_CWR_UNC3_MULT`.




	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 19 of 73

Table 5.2-1. POST2 Inputs for the Aerodynamic Dispersion Model

Input Symbol	Units	Stored Value	Definition
CAT_UNC1_MULT (UNC2 &UNC3)	nd	0.0	Axial force coefficient multiplicative uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CAT_UNC1_ADD (UNC2 &UNC3)	nd	0.0	Axial force coefficient additive uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CMQ_CWR_UNC1_ADD (UNC2 &UNC3)	nd	0.0	Pitch damping and yaw moment damping due to yaw rate additive dispersion values at cg for free molecular flight regime. Input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. Same value is applied to both dynamic derivatives. (Hypersonic and Supersonic flight regimes)
CMQ_CWR_UNC3_MULT	nd	0.0	Pitch damping and yaw moment damping due to yaw rate multiplicative dispersion values at cg in supersonic flight regime. Input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. Same value is applied to both dynamic derivatives.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 20 of 73

Input Symbol	Units	Stored Value	Definition
CMT_UNC1_M ULT (UNC2 &UNC3)	nd	0.0	Pitching moment coefficient multiplicative uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CMT_UNC1_ ADD (UNC2 &UNC3)	nd	0.0	Pitching moment coefficient additive uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CNT_UNC1_M ULT (UNC2 &UNC3)	nd	0.0	Normal force coefficient multiplicative uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CNT_UNC1_ ADD (UNC2 &UNC3)	nd	0.0	Normal force coefficient additive uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CW_UNC1_ MULT (UNC2 &UNC3)	nd	0.0	Yawing moment coefficient multiplicative uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value.. (Hypersonic and Supersonic flight regimes)

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 21 of 73

Input Symbol	Units	Stored Value	Definition
CW_UNC1_ADD (UNC2 &UNC3)	nd	0.0	Yawing moment coefficient additive uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CY_UNC1_MULT (UNC2 &UNC3)	nd	0.0	Side force coefficient multiplicative uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
CY_UNC1_ADD (UNC2 &UNC3)	nd	0.0	Side force coefficient additive uncertainty at cg in free molecular flight regime. For IDISP_AERO=1, input dispersion value as fraction of 3-sigma value; that is +1 is + 3-sigma, -1 is -3-sigma and 0.5 is half of the 3-sigma value. (Hypersonic and Supersonic flight regimes)
IDISP_AERO	Integer	0	Flag to generate dispersed aerodynamics. =0, use nominal (un-dispersed) aerodynamic coefficients =1, disperse aerodynamics at cg using inputs as fractions of \pm 3-sigma dispersions.
REDL_AEROU NCERT_j, j=1,50	nd	0.0	Array of aerodynamic uncertainties applied to nominal coefficients. Values input for standard or total aerodynamic uncertainties defined in this table also populate this array in the appropriate location.



	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 22 of 73

Table 5.2-2. POST2 Outputs from the Aerodynamic Dispersion Model

Output Symbol	Type/ Units	Definition
CA_NOM	nd	Nominal Axial force coefficient
CAT_NOM	nd	Nominal total Axial force coefficient
CLL_NOM	nd	Nominal rolling moment coefficient
CLLP_NOM	nd	Nominal roll moment damping coefficient due to roll rate
CLLR_NOM	nd	Nominal roll moment damping coefficient due to yaw rate
CM_NOM	nd	Nominal pitching moment coefficient
CMQ_NOM	nd	Nominal pitch moment damping coefficient due to pitch rate
CMT_NOM	nd	Nominal total pitching moment coefficient
CN_NOM	nd	Nominal Normal force coefficient
CNT_NOM	nd	Nominal total Normal force coefficient
CW_NOM	nd	Nominal yawing moment coefficient
CWP_NOM	nd	Nominal yaw moment damping coefficient due to roll rate
CWR_NOM	nd	Nominal yaw moment damping coefficient due to yaw rate
CY_NOM	nd	Nominal Side force coefficient
CA_DIS	nd	Dispersed Axial force coefficient
CAT_DIS	nd	Dispersed total Axial force coefficient
CLL_DIS	nd	Dispersed rolling moment coefficient
CLLP_DIS	nd	Dispersed roll moment damping coefficient due to roll rate
CLLR_DIS	nd	Dispersed roll moment damping coefficient due to yaw rate
CM_DIS	nd	Dispersed pitching moment coefficient
CMQ_DIS	nd	Dispersed pitch moment damping coefficient due to pitch rate
CMT_DIS	nd	Dispersed total pitching moment coefficient
CN_DIS	nd	Dispersed Normal force coefficient
CNT_DIS	nd	Dispersed total Normal force coefficient
CW_DIS	nd	Dispersed yawing moment coefficient
CWP_DIS	nd	Dispersed yaw moment damping coefficient due to roll rate
CWR_DIS	nd	Dispersed yaw moment damping coefficient due to yaw rate
CY_DIS	nd	Dispersed Side force coefficient
KNUDSEN	nd	Knudsen Number

5.3 Terminal Point Control Guidance Algorithm

A key element in developing rapid vehicle simulations is the vehicle guidance, navigation, and control (GN&C). The vehicle guidance (i.e., the "driver") takes input from the navigation system and user input targeting information to send signals to the flight control system that will guide the vehicle to its destination while remaining within the operating constraints. Several guidance algorithms were included in Phase 1. For this phase, the TPC guidance algorithm for aerocapture was added.


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 23 of 73

This guidance algorithm was implemented in order to examine vehicles that enter the atmosphere and aerocapture into orbit prior to EDL. The TPC guidance formulation was derived using the calculus of variations, which was the same methodology used to develop the Apollo Earth entry terminal phase guidance. A reference trajectory is used to determine the sensitivities of the final vehicle state to changes in the control at any point along the trajectory. For the TPC guidance, the control variable is the vehicle bank angle. The derivation differs from that of the Apollo terminal phase guidance in two significant ways. First, the boundary conditions for aerocapture are different from those for landing. Second, the aerocapture trajectory is not monotonic in altitude, so the simplification allowed by parameterization on altitude is not possible for aerocapture. The TPC guidance provides bank angle commands to reach a target orbit after a single atmospheric pass.

As in the entry guidance, the bank angle magnitude controls the in-plane lift. For aerocapture, the in-plane lift targets to the desired apoapsis while maintaining periapsis as high as possible. The bank angle sign is used for out-of-plane control of either inclination or wedge angle. The TPC algorithm uses a lateral corridor that is indexed on energy. Whenever the wedge or inclination angle error is outside the corridor, the bank angle sign is negated to drive it back into the corridor. There is no general procedure to determine the shape of the corridor. It is tuned to meet the needs of each individual mission. The TPC guidance algorithm theoretical development with equations is provided in references 6 and 7.

The process of tuning the TPC guidance starts with performing a reference 3-degree of freedom (DoF) trajectory. The reference trajectory is run using the input file PICKBANKNOM.INP. This input file can be set either to pick the required constant bank angle for a given entry flight path angle, or to pick the required entry flight path angle for a given bank angle profile. The latter allows a variable bank angle profile, typically a bank angle is used that is more lift up initially, more lift down at the end, and varies linearly with energy. The input file should be a nominal aerocapture trajectory except that the lift is in-plane and unguided (i.e., only an open loop bank angle profile is used). The input files PICKBANKSTP.INP and PICKBANKSHW.INP are the same except that one is steeper at entry, whereas the other is shallower. The amount of steepness or shallowness depends on the bank profile used. Targeting near, but not at the edge of the flyable corridor works best. Note that aerocapture trajectories can be sensitive. An initial bank angle estimate that results in the hyperbolic approach trajectory capturing into an elliptical orbit is required. A bank angle yielding in a flight path angle that is too steep at atmospheric interface results in the vehicle impacting into the planet's surface, and thus an acceptable derivative can not be generated. If the flight path angle is too shallow, then the vehicle skips out (i.e., does not achieve elliptical orbit) and no acceptable derivative can be generated. Fortunately, first order calculations can be used to converge on initial parameters.

After the reference trajectories are performed, form the .mat files with the extensions “_nom”, “_stp” and “_shw” for the nominal, steep and shallow trajectory cases, respectively. Then run

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 24 of 73

the script file “multimaster.m” in MATLAB™. For each of the three references, multimaster performs several functions:

- Retrieves the relevant information from the .mat file and saves it to a .dat file (pbnom.dat, pbstp.dat and pbshw.dat).
 - The following variables are needed in the print block: TIME, ALTITO, GAMIRV, VELR, RDOT, DRAGW, DENS, BNKANG, AEROCAP_DV, ENERGY, GCRAD, GXI, GYI, GZI, DECLN, LONG, ENRGDT, MALTA, WEIGHT, VELI, and CDW;
- Identifies the terminal costate boundary conditions based on the exit states and the target orbit;
- Integrates the costates backwards from the final condition and calculates the gains along the trajectory;
- Reduces the number of data points to minimize storage requirements;
- Plots the resulting data to check that the reduced data represents the original; and
- Writes the gains in the form of POST2 tables, including comments.

The gain tables are saved in the files TPC_gains_nom.dat, TPC_gains_shw.dat, and TPC_gains_stp.dat. The multimaster script calls: nomgenmulti.m, mgacdmulti.m, data_reduct.m, verify_plots.m, writepostinput_nom.m, writepostinput_stp.m, and writepostinput_shw.m. The script mgacdmulti.m is where the boundary conditions are calculated, the costates are integrated, and the gains are calculated. To run POST2 with the guidance in the loop, the input deck needs to include the three gain table files and the guidance setup file, tpc_setup.dat, and to have the guidance mode set to 15 (IGUID(14) = 15). Note that the parameters most adjusted to achieve a successful aerocapture are TPC_OVERKD, TPC_OVERKRD, and TPC_NSAMP. TPC_OVERKV is often set to zero.

POST2 inputs and outputs for the TPC guidance algorithm are provided in Tables 5.3-1 and 5.3-2, respectively. Note that all of the variables in the TPC guidance structure are prefixed with “tpc_”. Many of the input parameters are generated via MATLAB™ scripts and the nominal trajectory. Additionally, all of the gain tables are generated using MATLAB™ scripts. These gain table and input value files generated via the process must be included in the POST2 input file for the TPC guidance to function properly.



	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 25 of 73

Table 5.3-1. POST2 Inputs for the TPC Guidance Algorithm

Input Symbol	Units	Stored Value	Definition
IGUID(14)	integer	0	The TPC selection flag. =15, Use TPC guidance
TPC_REQ	m	0.0	Planet equatorial radius used in TPC
TPC_RPL	m	0.0	Planet polar radius used in TPC
TPC_WIE	rad/s	0.0	Planet rotation rate used in TPC
TPC_XMUE	m^3/s^2	0.0	Planet gravitational constant used in TPC
TPC_DT	s	0.0	TPC guidance call update time
TPC_GSTART, TPC_GSTOP	m/s^2	0.0, 0.0	Acceleration triggers to begin and end atmospheric phase of TPC guidance
TPC_TAU1, TPC_TAU2	s	0.0	TPC drag filter time constants
TPC_CDNOM	nd	0.0	Nominal vehicle drag coefficient used by TPC
TPC_SNOM	m^2	0.0	Nominal vehicle reference area used by TPC
TPC_DENSFACT	nd	0.0	TPC over-density factor applied to the atmospheric density
TPC_SCALEHEIGHT_NOM	km	0.0	Atmospheric scale height used to initialize the atmospheric density estimator in TPC
TPC_OVERKRD	nd	0.0	TPC overcontrol gain on drag acceleration
TPC_OVERKRD	nd	0.0	TPC overcontrol gain on radius rate of change
TPC_OVERKRV	nd	0.0	TPC overcontrol gain on velocity
TPC_DELTA_APO	m	0.0	Difference from the target apogee radius (TPC_RA_TARG) at which aeroshell is jettisoned (e.g., if =5000, aeroshell will jettison 5 km above target apoapsis)
TPC_RA_TARG	m	0.0	The target apogee radius at which aeroshell is jettisoned in TPC
TPC_RHO0			Atmospheric surface density used to initialize the atmospheric density estimator in TPC
TPC_NSAMPL	integer	0	Number of acceleration measurements used by the atmospheric density estimator in TPC
TPC_IAXIALFLAG	integer	0	Flag to determine which accelerations to use to estimate density in TPC = 2, use drag accelerations
TPC_MIN_RHO0, TPC_MAX_RHO0		0.0, 0.0	Lower and upper limits on the “sea-level” atmospheric density in the estimator in TPC

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 26 of 73

Input Symbol	Units	Stored Value	Definition
TPC_MIN_SCH, TPC_MAX_SCH	km	0.0, 0.0	Lower and upper limits on the atmospheric scale height in the estimator in TPC.
TPC_IFLG	integer	0	TPC mode flag. If =2, then the vehicle is forced to follow input reference bank profile.
TPC_BOTTOMETRIGGER	kg m ² /s ²	0.0	Energy trigger to change the roll reversal limits from TPC_BOTTOMHIE to TPC_BOTTOMLOE in TPC.
TPC_BOTTOMLOE, TPC_BOTTOMHIE	degree	0.0, 0.0	The low-energy and high-energy limits on bank reversal direction in TPC. If energy is more than TPC_BOTTOMETRIGGER and TPC_BOTTOMHIE=80, then reversals commanded when bank is less than 80 will go over the top, reversals commanded when bank is greater than 80 will go underneath. Set to 90 will give shortest distance.
TPC_IDBG	integer	0	TPC debug flag. Set =1 for additional output and generate an external file (fort.68) with more output.
TPC_LATFLAG	integer	0	TPC flag to select the lateral logic variable. = 0, inclination =1, wedge angle
TPC_BANKLIM	degree	0.0	Limits closest command to lift up or lift down (e.g., if tpc_banklim=10 no commands will be issued between 10 and -10, or above 170 or below -170).
TPC_NOMONLYFLAG	integer	0	TPC flag to use only the nominal reference profile.
TPC_TOPINCX, TPC_TOPINCY, TPC_BOTINCX, TPC_BOTINCY	kg m ² /s ²	0.0, 0.0, 0.0, 0.0	TPC tables used to control roll reversals. The X tables are energy values and the Y tables are wedge angle (or inclination) values. Whenever the vehicle exceeds the interpolated Y values, a reversal is commanded. The tables currently have length 6.
TPC_XMASS		0.0	Mass of the vehicle used in TPC
TPC_NRDT1_NOM, TPC_NRDT1_STP, TPC_NRDT1_SHW	integer	0.0, 0.0, 0.0	Array lengths for the gains in each of the sets of tables in TPC


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 27 of 73

Table 5.3-2. POST2 Outputs for the TPC Guidance Algorithm

Output Symbol	Type/ Units	Definition
TPC_BANKC	degree	Commanded Bank Angle
TPC_JETTISON_ON	nd	Flag indicating heatshield is to be jettisoned.
TPC_DIR	nd	Commanded bank maneuver direction = -2, bank through 180 degree (underneath) = -1, bank left = 0, go shortest distance = 1, bank right = 2, bank through 0 degree (over)

5.4 Aerobraking Mission Design Module

The program contains basic algorithms required to perform an aerobraking mission design simulation. It is a variation of the mission design software used for mission planning and aerobraking operations for Mars Odyssey and Reconnaissance Orbiter. The code allows for specific walk-in and operational maneuver strategies. Walk-in is the phase that transitions the spacecraft from its exo-atmospheric orbit to aerobraking without exceeding any of the spacecraft constraints (e.g., heat rate, temperature, dynamic pressure). This module is capable of monitoring eclipse times, lifetime constraints, and includes several options for aerobraking corridor control.

The input requires the use of repeating roving events for critical events such as atmosphere entry, periapsis, atmosphere exit, and apoapsis. Aerobraking missions with more than 1000 orbits will require multiple sets of critical events in the input file. The aerobraking mission design option is operating when $npc(9) = 3$ and $ab_flag=1$.

The general setup and procedure for using the aerobraking mission design module involves input files and scripts. Samples are provided in a `redlasim_aerobraking` directory. The contents of that directory include the following files:


TEST.INP is a standard input used with an aerobraking case. The event structure can be changed, but AB_EVENT_FLAG input must adhere to the following values:

AB_EVENT_FLAG = 30 in the atmosphere entry event;

AB_EVENT_FLAG = 50 in the periapsis event;

AB_EVENT_FLAG = 70 in the atmosphere exit event; and

AB_EVENT_FLAG = 90 in the apoapsis event.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 28 of 73

SPAWN2.pl is called from PREDICT_DV.f. This script submits the POST2 spawned job that runs while the original aerobraking simulation is suspended. Once completed, data is placed into appropriate files by the script for the original run to continue.

The RUNOUT4.PL script converts the variables saved in save_runout_variables.cpp to a MATLAB™ format once an input case has completed. This script creates the TEST_APO.mat and TEST_PER.mat files containing apoapsis and periapsis data, respectively.

The RUN_S_POST.PL script submits the jobs to the cluster for execution. Currently, only cluster execution is permitted (no local machine runs) due to the spawn case setup is for cluster use only at this time.


PLOT_AB.m is a sample plotting routine is also included for use with MATLAB™.

This script loads the TEST.mat, TEST_PER.mat, and TEST_APO.mat to plot the results of the aerobraking mission.

Test cases were developed which employ this aerobraking mission design module. The POST2 inputs, tables, and outputs for the aerobraking mission design module is provided in Tables 5.4-1, 5.4-2, and 5.4-3, respectively.

Table 5.4-1. POST2 Inputs for the Aerobraking Mission Design Module

Input Symbol	Units	Stored Value	Definition
npc(9)	integer	0	Propulsion type selection flag = 0, no thrust = 1, 2, Rocket or ramjet engine = 3, Instantaneous delta V option also Calculate Aerobraking DV = 4, Instantaneous delta V addition using current weight of propellant and specific impulse
AB_END_VALUE	decimal	0.0	Value of AB_END_CRITR_MODE to end simulation (i.e., if AB_END_CRITR_MODE = AB_MALTA, AB_END_VALUE = 450)
AB_END_FLAG	integer	0	Flag to end aerobraking simulation = 1 ends sim
AB_EVENT_FLG	integer	0	Aerobraking event flag
AB_FLAG	integer	0	Aerobraking Mission Design module activation flag. =1, module active

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 29 of 73

AB_DV_MODE	integer	0	ΔV magnitude selection flag 0 = NO_MANEUVER 1 = TABLE_LOOKUP 2 = PREDICT
SPAWN_END_VALUE	decimal	0.0	Value of SPAWN_END_CRITR_MODE to end spawned case.
AB_PREDICT_MODE	integer	0	Type of AB corridor 1 = ALT_CORRIDOR 2 = DENS_CORRIDOR 3 = HTRT_CORRIDOR 4 = DYNP_CORRIDOR
SPAWN_END_CRITR_MODE	integer	0	End of simulation criteria 1 = ALT 2 = ORB 3 = DAY
SPAWNP2_PATH	character	*132	Path to the SPAWNP2 to be used in the spawn cases. Note that the path to executable must be changed inside SPAWNP2 file.
LAST_WALKIN_ORB	integer	0	Value of the last walk in orbit.
LAST_WALKIN2_ORB	integer	0	Value of last walk in orbit after solar conjunction.

Table 5.4-2. POST2 Table Inputs for the Aerobraking Mission Design Module

Input Symbol	Units	Stored Value	Definition
LOWER_CORRT	decimal	0.0	Aerobraking lower corridor limit.
UPPER_CORRT	decimal	Stored Value	Aerobraking upper corridor limit.
DAYS_B4_ABMT	decimal	0.0	Days between aerobraking maneuvers.
CORR_TARGT	decimal	0.0	Aerobraking corridor target (i.e., 0.5 targets the middle of the corridor).
AWALKIN_DVT	decimal	0.0	Table of walk-in maneuvers as a function of orbit number.
WALKIN2_DVT	decimal	0.0	Table of second walk-in maneuvers as a function of orbit number.


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 30 of 73


Table 5.4-3. POST2 Outputs for the Aerobraking Mission Design Module

Output Symbol	Type/Units	Definition
ORBIT_NUM	nd	orbit revolution counter from first walkin. Orbit numbers start at periapsis
ORBIT2_NUM	nd	Orbit revolution counter from second walkin
AB_DAY_NUM	days	24-hr period counter (days of aerobraking)
DOWN_MANEUVER_DV	m/s	down maneuver indicator, equal to DV magnitude (abs value)
UP_MANEUVER_DV	m/s	up maneuver indicator, equal to DV magnitude (abs value)
SUM_MANEUVER_DV	m/s	Cumulative maneuver DV
AB_MANEUVER_COUNTER	nd	Orbit maneuver counter
SIM_START_TIME	s	Simulation start time
LOWER_CORR		Value of the lower aerobraking corridor. Type of corridor and units depend on AB_PREDICT_MODE
UPPER_CORR		Value of the upper aerobraking corridor. Type of corridor and units depend on AB_PREDICT_MODE
CORR_TARG		Target in the corridor. Type of corridor and units depend on AB_PREDICT_MODE
AB_MAX_DENS	kg/m ³	Maximum density on a particular aerobraking pass
AB_MAX_HTRT	w/cm ²	Maximum heat rate indicator (1/2 rho vel ³) on a particular aerobraking pass
AB_MAX_DYNP	Pa	Maximum dynamic pressure on a particular aerobraking pass
AB_ENTRY_LS	degree	Local Solar Longitude of a particular aerobraking pass entry

5.5 Earth Environment Models

5.5.1 2010 Earth-Global Reference Atmosphere Model


Earth-GRAM 2010 [ref. 2] uses an empirical database to provide atmospheric quantities (e.g., density, temperature, pressure, winds, and constituent concentrations) from the Earth's surface to orbital altitudes based on geographic location and time of year. This version is an update to the previously released Earth-GRAM 2007. Specific changes for this latest release include an updated lower atmosphere database (Global Upper Air Climatic Atlas replaced by monthly global climatology by the National Centers for Environmental Prediction, or NCEP), revised

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 31 of 73


boundary layer model, updated thermosphere models, scaling parameters for standard deviations modified, auxiliary profiles option, moisture corrections, and additional output parameters. The Earth-GRAM 2010 models have been implemented into POST2 and are accessible via the standard input (npc(5)=19). The POST2 inputs, tables, and outputs for Earth-GRAM 2010 are listed in Tables 5.5-1, 5.5-2, and 5.5-3, respectively.

Table 5.5-1. POST2 Inputs for the Earth-GRAM 2010 Model


Input Symbol	Type/ Units	Stored Value	Definition
NPC(5)	integer	2	The atmosphere model selection flag. =19, Earth GRAM 2010 model
NPC(6)	integer	0	Wind calculation flag = 4 Use GRAM determination of North-South, East-West, and Vertical winds
AP	decimal	0.0	Geomagnetic index
ATMPATH	character	“null”	Path name for “atmosdat” atmospheric data file
F10	solar flux units	0.0	Daily 10.7-cm flux
F10b	solar flux units	0.0	Mean 10.7-cm flux
IATMFL1	integer	1	The GRAM initialization flag. For first vehicle number that uses GRAM =0, Do not initialize GRAM =1, Initialize GRAM Required for first call to GRAM For other vehicles using GRAM = 0, Do not update initial random number – will maintain same atmosphere density variability profile as another vehicle (correlated atmospheres) = 1, Initialize atmosphere for new vehicle – no correlation between vehicle atmospheres

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 32 of 73

Input Symbol	Type/ Units	Stored Value	Definition
IATMFL2	integer	0	Random number seed – used when variable atmosphere mode (IATMFL3=1) is used
IATMFL3	integer	0	Perturbed atmosphere property options = 0, Mean density, temperature, pressure and speed of sound. = 1, Perturbed density, mean temperature and pressure, and perturbed speed of sound
IATMFL4	integer	0	Winds to use if NPC(6)=4 = 0, nominal GRAM winds = 1, nominal GRAM winds + perturbations
IAUXPROFILE	integer	0	Flag to indicate POST2 tables are used to define auxiliary profile. The auxiliary profile is used to override the default density, temperature, pressure and wind profiles. The use of this option does not allow for DUSTAU perturbations. = 0, Do not use POST2 tables for profile, use file define by PROFILE = 1, Use POST2 tables AUXTEMPT, AUXPREST, AUXDENST, AUXEWWINDT, and AUXNSWINDT to define auxiliary profile
IOPR	integer	0	random output option 1 = random output 2 = none
IDA	integer	0	Day of the month
NCEPHR	integer	5	Code for UT hour of day if NCEP climatology is used: 1=00 UT, 2=06UT, 3=12UT, 4=18UT, 5=all times of day combined, 0 to use NCEP time-of-day based on input UTC hour (ihro)
NCEPYR	integer	9008	y1y2 to use NCEP climatology for period-of-record (POR) from year y1 through year y2

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 33 of 73


Input Symbol	Type/ Units	Stored Value	Definition
IHRO	integer	0	Initial UTC (Greenwich) time hour
INITPERT	integer	0	Initial perturbations flag = 0 GRAM-derived random initial perturbations values = 1 User-selected initial perturbations
ITHERM	integer	0	Thermosphere model selection flag. = 1 MET (Jacchia) = 2 MSIS = 3 JB2006
IUS	integer	0	Unit number for atmosdat data
IUG	integer	0	Unit number for NCEP files
IURRA	integer	0	Unit number for Range Reference Atmosphere (RRA) data. = 0 none = xx actual unit number (recommend 42)
IYR	integer	0	4 digit or 2 digit year. If 2 digits are used IYR>56=19xx, IYR<57 =20xx
MINO	integer	0	Initial UTC (Greenwich) time minute
MN	integer	0	Month (1-12)
PATCHY	integer	0	Patchiness in perturbation model. = 0 no patchiness ≠ 0 patchiness
PROFILE	character	“null”	(Optional) auxiliary profile input file name. The auxiliary file is used to override the default density, temperature, pressure and wind profiles.
RDINIT	percent	0.0	Initial density perturbation model (percent of mean). Used if INITPERT = 1
RPINIT	percent	0.0	Initial pressure perturbation model (percent of mean). Used if INITPERT = 1
RPSCALE	decimal	1.0	Random density perturbation scale factor (0 = no perturbation) $0 \leq \text{RPSCALE} \leq 2$. If the table RPSCALET is used, then the table value will override this input.
RRAPATH	character	“null”	Directory for RRA data.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 34 of 73

Input Symbol	Type/ Units	Stored Value	Definition
RTINIT	percent	0.0	Initial temperature perturbation model (percent of mean). Used if INITPERT = 1
RUINIT	m/s	0.0	Initial eastward wind velocity perturbation model. Used if INITPERT = 1
RVINIT	m/s	0.0	Initial northward wind velocity perturbation model. Used if INITPERT = 1
RWINIT	m/s	0.0	Initial upward wind velocity perturbation model. Used if INITPERT = 1
SECO	decimal	0.0	Initial UTC (Greenwich) time minute
SITELIM	degree	0.0	Lat-Lon radius from RRA or PROFILE outside of which the RRA or PROFILE data are not used.
SITENEAR	degree	0.0	Lat-Lon radius from RRA or PROFILE inside of which the RRA or PROFILE data are used with 1.0 weighting factor. Between SITENEAR and SITELIM the weighting factor transitions from 1.0 to 0.0 smoothly.
S10	decimal	0.0	Extreme Ultraviolet (EUV) index (26-34 nm) scaled to F10 units, i.e., 0.0 corresponds to S10=F10
S10B	decimal	0.0	EUV 81-day center-averaged index scaled to F10B units (i.e., 0.0 corresponds to S10B=F10B)
XM10	decimal	0.0	MG2 index scaled to F10 units (i.e., 0.0 corresponds to XM10=F10)
XM10B	decimal	0.0	MG2 81-day center-average index scaled to F10 units (i.e., 0.0 corresponds to XM10B=F10)

Table 5.5-2. POST2 Tables for the Earth-GRAM 2010 Model


Input Symbol	Type/ Units	Stored Value	Definition
AUXDENST	kg/m ³	0.0	Auxiliary profile of natural log of atmospheric density. Used if IAUXPROFILE = 1.
AUXEWWINDT	m/s	0.0	Auxiliary profile of East/West wind. Used if IAUXPROFILE = 1.
AUXNSWINDT	m/s	0.0	Auxiliary profile of North/South wind. Used if IAUXPROFILE = 1.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 35 of 73

AUXPREST	N/m ²	0.0	Auxiliary profile of natural log of atmospheric pressure. Used if IAUXPROFILE = 1.
AUXTEMPT	degree K	0.0	Auxiliary profile atmospheric temperature. Used if IAUXPROFILE = 1.
RPSCALET	decimal	0.0	Random density perturbation scale factor (0 = no perturbation) $0 \leq \text{RPSCALET} \leq 2$. If the table RPSCALET is used, then the table value will override the RPSCALE input.

Table 5.5-3. POST2 Outputs for the Earth-GRAM 2010 Model

Output Symbol	Type/ Units	Definition
ARMOLE	percent	Percentage of AR by volume in atmosphere.
ATEM	degree K	Atmospheric temperature.
AUXDENS	kg/m ³	Auxiliary profile density. Calculated if IAUXPROFILE = 1.
AUXEWWIND	m/s	Auxiliary profile East/West wind. Calculated if IAUXPROFILE = 1.
AUXNSWIND	m/s	Auxiliary profile North/South wind. Calculated if IAUXPROFILE = 1.
AUXPRES	N/m ²	Auxiliary profile pressure. Calculated if IAUXPROFILE = 1.
AUXTEMP	degree K	Auxiliary profile temperature. Used if IAUXPROFILE = 1.
CH4MOLE	percent	Percentage of CH ₄ by volume in atmosphere.
COMOLE	percent	Percentage of CO by volume in atmosphere.
CO2MOLE	percent	Percentage of CO ₂ by volume in atmosphere.
CS	m/s	Speed of sound.
DENS	kg/m ³	Atmospheric density. $\text{DENS} = \text{DENS} * \text{GENTAB}(\text{DENKT})$
DENSMEAN	kg/m ³	Mean atmospheric density.
DENSM3S	kg/m ³	3 sigma low atmospheric density.
DENSP3S	kg/m ³	3 sigma high atmospheric density.
DENSRAT	decimal	Ratio of density to mean density
DENS76STAND	kg/m ³	1976 standard atmospheric density.
EWINDMEAN	m/s	Mean East/West wind velocity. Positive to the East.
HEMOLE	percent	Percentage of HE by volume in atmosphere.
HMOLE	percent	Percentage of H by volume in atmosphere.
H2OMOLE	percent	Percentage of H ₂ O by volume in atmosphere.
MOLWEIGHT	decimal	Molecular weight of atmosphere
NMOLE	percent	Percentage of N by volume in atmosphere.
N2MOLE	percent	Percentage of N ₂ by volume in atmosphere.
N2OMOLE	percent	Percentage of N ₂ O by volume in atmosphere.
NSWINDMEAN	m/s	Mean North/South wind velocity. Positive to the North.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 36 of 73


Output Symbol	Type/ Units	Definition
OMOLE	percent	Percentage of O by volume in atmosphere.
O2MOLE	percent	Percentage of O ₂ by volume in atmosphere.
O3MOLE	percent	Percentage of O ₃ by volume in atmosphere.
PRES	N/m ²	Atmospheric pressure.
PRESMEAN	N/m ²	Mean atmospheric pressure.
PRES76STAND	N/m ²	1976 standard atmospheric pressure.
TEMPMEAN	degree K	Mean atmospheric temperature.
TEMP76STAND	degree K	1976 standard atmospheric temperature.
VERTWINDMEAN	m/s	Mean vertical wind velocity. Positive down.
WINDEW	m/s	East/West wind velocity. Positive East. Used if NPC(6) =4
WINDNS	m/s	North/South wind velocity. Positive North. Used if NPC(6) =4
WINDV	m/s	Vertical wind velocity. Positive down. Used if NPC(6) =4
ZHPRES	km	Atmospheric pressure scale height.
ZHRHO	km	Atmospheric density scale height.

5.5.2 Earth Gravity Model 96

The EGM96 model was formatted for use with the POST2 gravity model. The EGM96 (complete to degree and order 360) was generated by NASA and other Government agencies. This model is being used with various NASA simulations evaluating currently proposed Earth entry systems. Various data sets including satellite tracking and altimetry were used to generate this model. Standard POST2 inputs and outputs for gravity models are used; the spherical harmonic gravity model is invoked in POST2 using NPC(16)=7, and the file containing the sectoral and tesseral data is identified in the input (including system directory path) using the POST2 variable GRAVDATA. A test input file was generated using EGM96.

5.6 Attitude Control Models

In an effort to balance faster executing and more easily developed 3 DoF simulations (versus the slower, higher fidelity 6 DoF simulations), control models that address vehicle attitude change are used. A pseudo controller model was included in Phase 1 which used a method of emulating a 6 DoF attitude control system for bank angle modulation. Additionally in Phase 1, the natural aerodynamic trim points in pitch and yaw to determine the angle of attack (AOA) and sideslip angle that the vehicle orients to at any time during the atmospheric entry were included to provide 3-axis vehicle control. For Phase 2, another method of 3-axis vehicle control based on the bank angle pseudo-controller was developed. Also, a utility function to provide a second order actuator model for use with models within POST2 (e.g., engine gimbal or sensor slewing motion) is included. Both of these models were incorporated in the simulation and their implementations are described in the following subsections.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 37 of 73

5.6.1 Pseudo-Controller for 3-axis Attitude Control

The program contains a pseudo-controller module that emulates the performance of a feedback control system. The three-axis pseudo-controller (3APC) is a generalization of the bank angle pseudo-controller (BPC) and allows a 3 DoF simulation to model some of the dynamic attitude behavior of a 6 DoF trajectory. The 3APC can be used for 3-axis attitude control of any 3 DoF vehicle trajectory. However, if the vehicle being modeled uses bank angle to modulate lift, then the BPC will be a more straightforward implementation. Note that this module does not solve the rotational equations of motion and does not require moments of inertia. Also, the 3APC module treats each axis independently and should not be used for systems with high angular rates or other significant cross-axis coupling.


The 3APC expects three commanded angles from the guidance system: roll, pitch and yaw. The 3APC attempts to achieve these commanded angles by applying angular accelerations under the constraint of second-order dynamics. Maximum allowed values for the angular acceleration and angular rate are given as inputs:

$$\phi = \phi_0 + \dot{\phi} dt + \frac{1}{2} \ddot{\phi} d t^2$$

The 3APC module maintains internal states of angle and angle rate for each axis, which are integrated within the model, not using the POST2 generalized integration procedure. Angular rate and acceleration can be adjusted based on current residual angle error (i.e., difference between commanded and actual angle) and when the actual angle overshoots the command.

Several options are available to model behavior of the 3APC when the vehicle overshoots the command. Since the 3APC bases the angular acceleration, it uses on the current angle and the commanded angle, overshoots should only occur due to a change in the command. The “normal” option is to use all of the vehicles deceleration capability to reach the commanded angle. The “no overshoot” option allows the vehicle to exceed the maximum angular acceleration when the vehicle passes the commanded angle in order to prevent an overshoot. The “no wrong way” option ensures that the angle only moves toward the commanded angle. If the command changes in such a way that the current angular rate moves the vehicle away from the current command, the “no wrong way” option forces the angle to the new command. Finally, the “perfect” controller option instantly sets the angle to the command and the angular rate to zero.

The angle command and the angle direction command are provided from the guidance or the input deck. There are several options for angle direction and for how the controller handles overshoots. The angle direction flag can be set to: under, left, shortest, right, or over. “Under” means the vehicle will rotate toward an angle of 180 degrees. “Left” means the vehicle will rotate to the left. “Shortest” means bank in the direction that minimizes the absolute angle error. “Right” means the vehicle will rotate to the right, and “over” means through an angle of 0 degrees.


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 38 of 73

No limitations are placed on the size of the maximum acceleration or rate that are input. If large enough values are used, then the 3APC will operate as a near instantaneous angle controller. Once input, these limits are used to limit the maximum values in an absolute value sense (i.e., \pm maximum value are the limit boundaries used).

Inputs and outputs for the POST2 implementation of the 3APC are provided in Tables 5.6-1 and 5.6-2, respectively. To use the 3APC, the relative euler angles steering option must be used (IGUID(1)=2), and the angle polynomial with constant term from input (IGUID(4) =1 or IGUID(9) =3*1, depending on value of IGUID(2).

Table 5.6-1. POST2 Inputs for the 3APC Model


Input Symbol	Units	Stored Value	Definition
CTRL_ROL_CMD, CTRL_PIT_CMD, CTRL_YAW_CMD	degree	0.0	Commanded angle about the X (ROL), Y (PIT), and Z (YAW) body axes.
CTRL_ROL_DT, CTRL_PIT_DT, CTRL_YAW_DT	seconds	0.0	Pseudo-controller update cycle time for the X (ROL), Y (PIT), and Z (YAW) body axis controller.
CTRL_ROL_IDIR_FLAG, CTRL_PIT_IDIR_FLAG, CTRL_YAW_IDIR_FLAG	integer	0	Flag to control maneuver direction for the X (ROL), Y (PIT), and Z (YAW) body axis controller = -2, bank through 180 degree (underneath) = -1, bank left = 0, go shortest distance = 1, bank right = 2, bank through 0 degree (over)
CTRL_ROL_IPSEUDO_FLAG, CTRL_PIT_IPSEUDO_FLAG, CTRL_YAW_IPSEUDO_FLAG	integer	0	The Pseudo-controller selection flag for the X (ROL), Y (PIT), and Z (YAW) body axis controller. =1, Use Pseudo-Controller
CTRL_ROL_MAXACCEL, CTRL_PIT_MAXACCEL, CTRL_YAW_MAXACCEL	deg/s ²	5.0	Maximum angular acceleration used by the pseudo-controller flag for the X (ROL), Y (PIT), and Z (YAW) body axis
CTRL_ROL_MAXRATE, CTRL_PIT_MAXRATE, CTRL_YAW_MAXRATE	deg/s	20.0	Maximum angular rate used by the pseudo-controller flag for the X (ROL), Y (PIT), and Z (YAW) body axis

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 39 of 73

Input Symbol	Units	Stored Value	Definition
CTRL_ROL_OSMODE, CTRL_PIT_OSMODE, CTRL_YAW_OSMODE	integer	0	Overshoot mode selection flag for the X (ROL), Y (PIT), and Z (YAW) body axis = 0, normal = 1, no overshoot = 2, no wrong way = -1, perfect
IGUID(1)	integer	0	Type of steering (guidance) desired. =0, AOA, sideslip, and bank. Also input values for IGUID(2) and IGUID(3) or IGUID(6), IGUID(7), and IGUID(8).
IGUID(3)	integer	0	A flag to specify the steering option when commanding all channels simultaneously using aerodynamic AOA, sideslip, and bank angle. Must use option 1 with BPC. = 1, Command AOA, sideslip, and bank as third order polynomials with the values of the constant terms of the polynomials are the input values.
IGUID(8)	integer	0	Steering option flag when using separate channel for bank angle. Must use option 1 with BPC. = 1, Command bank angle as third order polynomials except that the constant terms of the polynomials are the input values.

Table 5.6-2. POST2 Outputs for the 3APC Model

Output Symbol	Type/ Units	Definition
CTRL_ROL_CMD, CTRL_PIT_CMD, CTRL_YAW_CMD	degree	Angle command for the X (ROL), Y (PIT), and Z (YAW) body axis
CTRL_ROL_DOT, CTRL_PIT_DOT, CTRL_YAW_DOT	deg/s	Angular rate about the X (ROL), Y (PIT), and Z (YAW) body axis
CTRL_ROL_PC1, CTRL_PIT_PC1, CTRL_YAW_PC1	degree	X (ROL), Y (PIT), and Z (YAW) body axis angle polynomial first coefficient – generated by pseudo-controller

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 40 of 73

5.6.2 Second Order Actuator Model

A general purpose second order actuator model has been developed for use internally with POST2 models (e.g., engine gimbal or sensor slewing). The routine, actu8, models second order dynamic system of the form:

$$\frac{d^2x}{dt^2} + 2\zeta\omega \frac{dx}{dt} + \omega^2 x = u \quad (5.6-1)$$

□ where x is the displacement of the system; ζ is the damping; ω is the natural frequency; and u is the input to the system. The model expects to be called at every time step and integrates the internal state variables across a time interval that must be supplied. The user provides the actuator damping and natural frequency, the update cycle time, the position and rate limits, and the current position command.

Inputs to the routine are:

- System initial position
- System initial rate
- Time step
- Actuator damping ratio
- Actuator natural frequency
- Commanded position
- Plus and minus rate limits
- Plus and minus displacement limits

Outputs from the system are:


- Final position after time step
- Final rate after time step

Note that the final position and rate from one call are used as the initial position and rate for the next call. This feature allows the same actuator model code to be used for multiple actuators, at the expense of external storage for the state variables.

6.0 Observation and NESC Recommendation

The following observation and NESC recommendation relate to technical aspects of this assessment:

- O-1.** The validated and documented subsystem models, test cases, and scripts developed during Phases 1 and 2 of this assessment, and added to the POST2 simulation framework, have increased the Agency's ability to rapidly evaluate EDL characteristics in systems

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 41 of 73

analysis studies, preliminary design, mission development and execution, and time-critical assessments.


- R-1.** The custodians of the POST2 simulation framework, the Atmospheric Flight and Entry Systems Branch at LaRC, should retain and maintain the REDLAS models, test cases, and scripts for use by current and future programs. *(O-1)*

7.0 Other Deliverables

There are no other deliverables.

8.0 Definition of Terms

Corrective Actions	Changes to design processes, work instructions, workmanship practices, training, inspections, tests, procedures, specifications, drawings, tools, equipment, facilities, resources, or material that result in preventing, minimizing, or limiting the potential for recurrence of a problem.
Finding	A conclusion based on facts established by the investigating authority.
Lessons Learned	Knowledge or understanding gained by experience. The experience may be positive, as in a successful test or mission, or negative, as in a mishap or failure. A lesson must be significant in that it has real or assumed impact on operations; valid in that it is factually and technically correct; and applicable in that it identifies a specific design, process, or decision that reduces or limits the potential for failures and mishaps, or reinforces a positive result.
Observation	A factor, event, or circumstance identified during the assessment that did not contribute to the problem, but if left uncorrected has the potential to cause a mishap, injury, or increase the severity should a mishap occur. Alternatively, an observation could be a positive acknowledgement of a Center/Program/Project/Organization's operational structure, tools, and/or support provided.
Problem	The subject of the independent technical assessment.
Proximate Cause	The event(s) that occurred, including any condition(s) that existed immediately before the undesired outcome, directly resulted in its occurrence and, if eliminated or modified, would have prevented the undesired outcome.


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 42 of 73

Recommendation An action identified by the NESC to correct a root cause or deficiency identified during the investigation. The recommendations may be used by the responsible Center/Program/Project/Organization in the preparation of a corrective action plan.

Root Cause One of multiple factors (events, conditions, or organizational factors) that contributed to or created the proximate cause and subsequent undesired outcome and, if eliminated or modified, would have prevented the undesired outcome. Typically, multiple root causes contribute to an undesired outcome.

9.0 Acronyms List

3APC	Three Axis Pseudo-Controller
AMA	Analytical Mechanics Associates, Inc.
AOA	Angle-Of-Attack
ARC	Ames Research Center
ATK	Alliant Techsystems, Inc.
BPC	Bank Angle Pseudo-Controller
CFD	Computational Fluid Dynamics
cg	Center of Gravity
DFRC	Dryden Flight Research Center
DoF	Degree of Freedom
EDL	Entry, Descent, and Landing
EDL-SA	EDL Systems Analysis
EGM	Earth Gravity Model
EKF	Extended Kalman Filter
EUV	Extreme Ultraviolet
GN&C	Guidance, Navigation, and Control
GRAM	Global Reference Atmosphere Model
GSFC	Goddard Space Flight Center
IMU	Inertial Measurement Unit
JSC	Johnson Space Center
KSC	Kennedy Space Center
LaRC	Langley Research Center
LOX	Liquid Oxygen
mEKF	multimode EKF
MER	Mars Exploration Rover
MSFC	Marshall Space Flight Center
MSL	Mars Science Laboratory

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 43 of 73


N ₂	nitrogen
NCEP	National Centers for Environmental Prediction
nd	non-dimensional
NESC	NASA Engineering and Safety Center
NIA	National Institute for Aerospace
NRB	NESC Review Board
O ₂	oxygen
O ₃	the molecular formula for ozone
POR	period-of-record
POST2	Program to Optimize Simulated Trajectories II
REDLAS	Rapid EDL Analysis Simulation
RRA	Range Reference Atmosphere
TPC	Terminal Point Control

10.0 References

1. Murri, D. G.; Striepe, S.A. Dr.; Powell, R.W. NESC Report Assessment #09-00530, Phase 1: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis, NESC-RP-09-00530. NASA-TM-2010-216867. December 17, 2009.
2. Justus, C.G. and Leslie, F.W., "The NASA MSFC Earth Global Reference Atmospheric Model – 2007 Version," NASA TM 2008-215581. 2010 Version in press.
3. Braun, Robert D., Mitcheltree, Robert A., and Cheatwood, F. McNeil: "Mars Microprobe Entry-to-Impact Analysis," Journal of Spacecraft and Rockets, Vol. 36, No. 3, May-June 1999.
4. Desai, Prasun N., Schoenenberger, Mark, and Cheatwood, F. McNeil: "Mars Exploration Rover Six-Degree-of-Freedom Entry Trajectory Analysis," Journal of Spacecraft and Rockets, Vol. 43, No. 5, September-October 2006.
5. Desai, Prasun N. and Cheatwood, F. McNeil: "Entry Dispersion Analysis for the Genesis Sample Return Capsule," Journal of Spacecraft and Rockets, Vol. 38, No. 3, May-June 2001.
6. Ro, T.U.; Queen, E.M.; and Striepe, S.A.: "Martian Aerocapture Terminal Point Guidance: A Reference Path Optimization Study," Presented at the 1999 Flight Mechanics Symposium at NASA Goddard Spaceflight Center, Greenbelt, MD, May 1999, NASA/CP-1999-209235, pp. 85-99.
7. Ro, T. U., Queen, E. M., "Mars Aerocapture Terminal Point Guidance," Paper No. 98-4571, AIAA Atmospheric Flight Mechanics Conference, Boston, MA, August, 1998.


Volume II: Appendices

Appendix A. Multi-Mode Extended Kalman Filter

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 44 of 73

Appendix A. Multi-Mode Extended Kalman Filter

**Work in this Appendix was performed under contract NNL07AA00B,
Task Order NNL07AM00T**

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 45 of 73


NASA Engineering and Safety Center
Simulation Framework for Rapid EDL
MULTI-MODE EXTENDED KALMAN FILTER

Dr. Robert H. Bishop




V1.1
JANUARY 24, 2011



	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 46 of 73


Document Change History

Date	Comments	Pages
11 – 17 – 10	Original Version	1 – 29
1 – 24 – 11	With NASA Revisions	1 – 29

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 47 of 73

Contents

1	Introduction	4
2	Extended Kalman Filtering Basics	6
2.1	Architecture	6
2.2	Propagation	7
2.3	Update	9
3	Mathematical Model of the Flight Dynamics	10
3.1	Dynamical Model in Continuous-time	10
3.2	Inertial Measurement Units	11
3.2.1	Accumulated Integrated Accelerometers and Gyros	13
3.3	Dynamical Model in Discrete-time	14
3.4	State Estimation Errors	15
3.4.1	State Estimation Error Propagation	15
3.4.2	State Estimation Error Covariance	17
4	Multi-mode Extended Kalman Filter	18
4.1	Mathematical model	18
4.2	mEKF Hierarchical Structure	18
5	Sensor Modeling	22
5.1	Sensor hierarchical structure	22
5.2	Inertial Measurement Unit	25
5.2.1	IMU Sensor Hierarchical Structure	25
5.3	Altimeter	26
5.3.1	Mathematical model	26
5.3.2	Altimeter Sensor Hierarchical Structure	26
5.4	Velocimeter	26
5.4.1	Mathematical model	26
5.4.2	Velocimeter Sensor Hierarchical Structure	27
5.5	Attitude Camera	27
5.5.1	Mathematical model	27
5.5.2	Attitude Camera Sensor Hierarchical Structure	28

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 48 of 73

1 Introduction

The main objectives of the project are:

- Develop a multimode Kalman navigation filter (mKNF) architecture for application to entry, descent, and landing (EDL) using a standard set of measurements including altimeters, velocimeters, star trackers, and inertial measurement units. The main computational unit of the navigation architecture is the extended Kalman filter algorithm employing a dual-inertial state implementation with integrated position, velocity, and attitude. The filter will be readily re-configurable to accommodate differing mission scenarios (such as lunar entry versus Mars entry).
- Develop sensor models as required to support the mKNF and requisite mathematical specifications documentation. The error models will include systematic and random errors applicable to the individual sensors, including biases, misalignments, scaling factors, and random noise.
- Develop and execute test scripts to verify and validate the mKNF. This includes specifying test procedures, creating run matrices, reviewing the simulation outputs, creating run summaries, and commenting on the results.

To accomplish the tasks, algorithms were packaged in compact units of code with input/output tightly controlled using data structures. Each individual unit of code would be entirely re-entrant; hence the use of global variables and persistent variables is disallowed. Two main hierarchical structures were developed: the Multi-mode EKF Hierarchical Structure (MHS) and the Sensor Hierarchical Structure (SHS). The MHS shown in Figure 1 enables the mEKF to be a reusable unit from multiple calling programs. All of the inputs and outputs to the mEKF shown in Figure 1 are data structures with definitions described in Section 4.

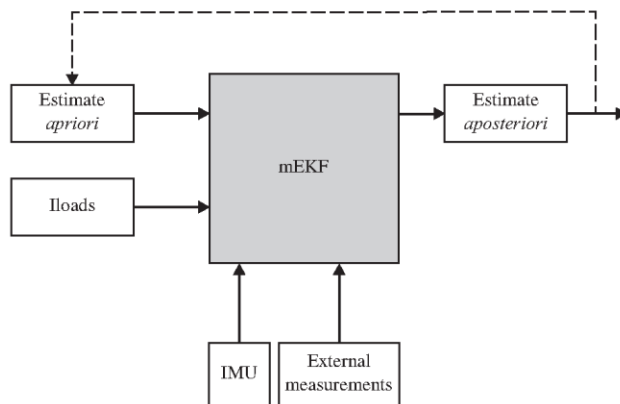



Figure 1: Multi-mode EKF Hierarchical Structure

Similarly, a sensor hierarchical structure (SHS), shown in Figure 2, is proposed to represent a convenient vehicle for representing the sensors required for EDL. The SHS is comprised of four major substructures: Universe, Nucleus, Identity, and Errors.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 49 of 73

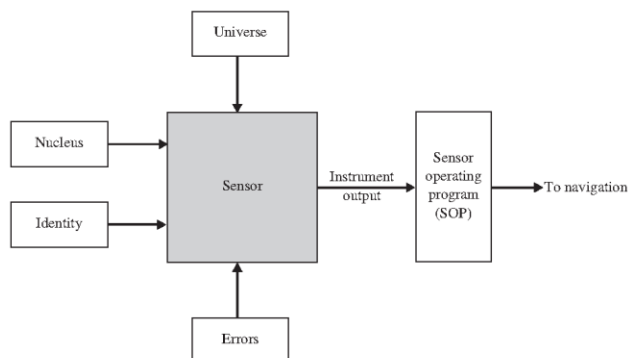



Figure 2: Sensor Hierarchical Structure

The Universe structure provides the parameters required by the filter that describe the external environment. The Nucleus structure describes the key elements of the sensor around which the model is constructed including, for example, the rate (internal), output rate, on/off events, and special processing (hardware specific) parameters. Currently, the Nucleus structure models the velocimeter, spherical altimeter, attitude camera, and IMU. The Identity structure describes the physical aspects of the sensor such as, for example, the spacecraft sensor layout (sensor locations, sensor reference frame orientations, etc.), and beam locations (for the velocimeter and altimeter). The Errors structure includes the random and systematic errors associated with each sensor, such as biases, noise strengths, misalignments, and time constants. Details of the SHS are provided in Section 3 for the IMU and in Section 5 for the altimeter, velocimeter, and attitude camera.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 50 of 73

2 Extended Kalman Filtering Basics

2.1 Architecture

An extended Kalman filter (EKF) is a model-based algorithm. The ability of the EKF to provide accurate estimates of the state of the system depends explicitly on the quality of the model. In the development of the EKF we assume a nonlinear system model is assumed to be of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) + \mathbf{M}(t)\mathbf{w}(t), \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state of the system, $\mathbf{f}(\mathbf{x}(t), t) \in \mathbb{R}^n$ is a sufficiently differentiable nonlinear system model, and $\mathbf{w}(t) \in \mathbb{R}^p$ is the process noise, which is assumed to have the properties

$$\mathbb{E}\{\mathbf{w}(t)\} = 0 \quad \text{and} \quad \mathbb{E}\{\mathbf{w}(t)\mathbf{w}^T(\tau)\} = \mathbf{Q}_{spec}\delta(t - \tau),$$

where $\mathbb{E}\{\cdot\}$ is the expectation operator and $\delta(t - \tau)$ is the Dirac delta function. The process noise is assumed to be a zero-mean, white-noise process with a constant-power spectral density given by $\mathbf{Q}_{spec} \in \mathbb{R}^{p \times p}$, which is an input to the EKF. Furthermore, $\mathbf{M}(t) \in \mathbb{R}^{n \times p}$ is the process noise mapping matrix.

The nonlinear measurement model is assumed to be of the form

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

where the subscript “ k ” denotes a discrete time measurement at time $t = t_k$. Furthermore, $\mathbf{h}_k(\mathbf{x}_k) \in \mathbb{R}^m$ is a sufficiently differentiable nonlinear measurement model evaluated at the state $\mathbf{x}_k = \mathbf{x}(t_k)$, and $\mathbf{v}_k \in \mathbb{R}^m$ is the measurement noise, assumed to be a white noise sequence with

$$\mathbb{E}\{\mathbf{v}_k\} = 0 \quad \text{and} \quad \mathbb{E}\{\mathbf{v}_k\mathbf{v}_{k'}^T\} = \mathbf{R}_k\delta_{kk'},$$

where k and k' are two different discrete measurement times and $\delta_{kk'}$ is the Kronecker delta. Additionally, it is assumed that the process noise and the measurement noise are not correlated in time, or, that

$$\mathbb{E}\{\mathbf{w}(t)\mathbf{v}_k^T\} = 0 \quad \forall t, t_k.$$

Letting $\mathbf{x}(t)$ denote the true state at some time t and that the estimate of the true state is denoted by $\hat{\mathbf{x}}(t)$, define the estimation error as

$$\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t). \quad (3)$$


By design, the EKF algorithm creates an unbiased state estimation, so that

$$\mathbb{E}\{\mathbf{e}(t)\} = 0 \quad \forall t. \quad (4)$$

This requires that the initial state estimate $\mathbf{x}(t_0)$ is such that the initial estimation error is zero mean. This is accomplished by setting $\mathbf{x}(t_0) = \mathbb{E}\{\mathbf{x}(t_0)\}$. We also define the state estimation error covariance as

$$\mathbb{E}\{\mathbf{e}(t)\mathbf{e}^T(t)\} = \mathbf{P}(t) \quad \forall t. \quad (5)$$

The EKF is a recursive algorithm comprised of two main phases: propagation and update. Figure 3 illustrates the propagation phase and the update phase. The state estimate and the state estimation error covariance are propagated forward to the time a discrete measurement becomes available at time t_k . At this time, the measurement is used to update both the state estimate and the state estimation error covariance. The values immediately preceding an update are called as *a priori* and denoted with a superscript “ $-$ ”, while the values immediately following an update are called as *a posteriori* and denoted with a superscript “ $+$ ”. The state estimation error covariance is a measure of uncertainty in the state estimate, so we expect it to be reduced (on average) as measurements are processed during the update phase. In the mEKF, the external sensors (currently) include the attitude camera, spherical altimeter, and velocimeter.

	<h1 style="text-align: center;">NASA Engineering and Safety Center</h1> <h2 style="text-align: center;">Technical Assessment Report</h2>	Document #: NESC-RP-09-00530	Version: 1.0
Title: <h3 style="text-align: center;">Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis</h3>			Page #: 51 of 73

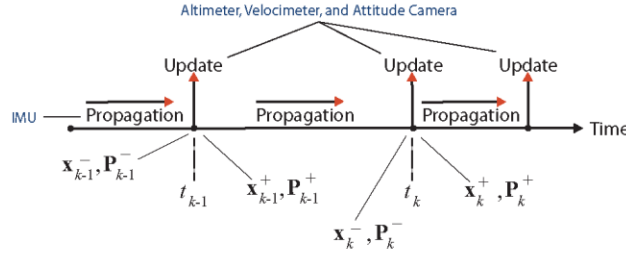


Figure 3: EKF Timeline.

It is during the update phase that the model of the sensor plays a key role. During the propagation phase there are no external measurements, so we expect (on average) that the state estimation error covariance increases (in other words, our uncertainty in the state estimate increases). During the propagation phase, the model of the dynamics of the system plays a key role. The inertial measurement unit (IMU) assists in the propagation of the state estimate by measuring and accounting for the nongravitational accelerations acting on the system.

2.2 Propagation

Taking the expected value of Eq. (1) yields

$$\dot{\hat{\mathbf{x}}}(t) = \mathbb{E} \{ \dot{\mathbf{x}}(t) \} = \mathbb{E} \{ \mathbf{f}(\mathbf{x}(t), t) \} + \mathbf{M}(t) \mathbb{E} \{ \mathbf{w}(t) \}.$$

Expanding $\mathbf{f}(\mathbf{x}(t), t)$ in a Taylor series expansion about $\hat{\mathbf{x}}(t)$ and neglecting higher-order terms yields

$$\dot{\hat{\mathbf{x}}}(t) = \mathbb{E} \{ \mathbf{f}(\hat{\mathbf{x}}(t), t) + \mathbf{F}(\hat{\mathbf{x}}(t), t) (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \} + \mathbf{M}(t) \mathbb{E} \{ \mathbf{w}(t) \}, \quad (6)$$

where

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \left. \frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}(t) = \hat{\mathbf{x}}(t)}.$$

Eq. (6) can be simplified as

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), t) + \mathbf{F}(\hat{\mathbf{x}}(t), t) \mathbb{E} \{ (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \} + \mathbf{M}(t) \mathbb{E} \{ \mathbf{w}(t) \},$$

Noting that the EKF is an unbiased estimator ($\mathbb{E} \{ (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \} = 0$) and that the process noise is assumed to be zero-mean ($\mathbb{E} \{ \mathbf{w}(t) \} = 0$), the differential equation for the estimated state can be written as


$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), t). \quad (7)$$

Using the estimation error in Eq. (3), and taking the time derivative yields

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{x}}(t) - \dot{\hat{\mathbf{x}}}(t). \quad (8)$$

Substituting $\dot{\mathbf{x}}(t)$ from Eq. (1) and $\dot{\hat{\mathbf{x}}}(t)$ from Eq. (7) into Eq. (8) it follows that

$$\dot{\mathbf{e}}(t) = \mathbf{f}(\mathbf{x}(t), t) + \mathbf{M}(t) \mathbf{w}(t) - \mathbf{f}(\hat{\mathbf{x}}(t), t).$$

	<h1 style="text-align: center;">NASA Engineering and Safety Center</h1> <h2 style="text-align: center;">Technical Assessment Report</h2>	Document #: NESC-RP-09-00530	Version: 1.0
Title: <h3 style="text-align: center;">Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis</h3>			Page #: 52 of 73

Using a Taylor series expansion of the true dynamics, represented by $\mathbf{f}(\mathbf{x}(t), t)$, about the estimated state of the system, represented by $\hat{\mathbf{x}}(t)$, and neglecting the higher-order terms yields

$$\dot{\mathbf{e}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t) (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) + \mathbf{M}(t)\mathbf{w}(t).$$

Therefore, making use of the estimation error as defined in Eq. (3) and, after algebraic manipulation, we obtain the estimation error differential equation

$$\dot{\mathbf{e}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\mathbf{e}(t) + \mathbf{M}(t)\mathbf{w}(t). \quad (9)$$

The solution of Eq. (9) is

$$\mathbf{e}(t) = \Phi(t, t_0)\mathbf{e}(t_0) + \int_{t_0}^t \Phi(t, \tau)\mathbf{M}(\tau)\mathbf{w}(\tau)d\tau,$$

for all $t \geq t_0$, where the state transition matrix $\Phi(t, t_0)$ corresponds to $\mathbf{F}(\hat{\mathbf{x}}(t), t)$, and satisfies the matrix differential equation

$$\dot{\Phi}(t, t_0) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\Phi(t, t_0),$$

subject to the initial condition

$$\Phi(t_0, t_0) = \mathbf{I}_{n \times n}.$$

From the definition of the estimation error covariance in Eq. (4), it follows that

$$\mathbf{P}(t) = \Phi(t, t_0)\mathbf{P}(t_0)\Phi^T(t, t_0) + \int_{t_0}^t \Phi(t, \tau)\mathbf{M}(\tau)\mathbf{Q}_{spec}(\tau)\mathbf{M}^T(\tau)\Phi^T(t, \tau)d\tau.$$

Define the matrix $\mathbf{Q}(t) \in \mathbb{R}^{n \times n}$ as

$$\mathbf{Q}(t) = \int_{t_0}^t \Phi(t, \tau)\mathbf{M}(\tau)\mathbf{Q}_{spec}(\tau)\mathbf{M}^T(\tau)\Phi^T(t, \tau)d\tau.$$

Taking the time derivative of $\mathbf{Q}(t)$ yields

$$\dot{\mathbf{Q}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\mathbf{Q}(t) + \mathbf{Q}(t)\mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{M}(t)\mathbf{Q}_{spec}(t)\mathbf{M}^T(t),$$

with the initial condition

$$\mathbf{Q}(t_0) = \mathbf{0}_{n \times n}.$$

To propagate the state estimate and the state estimation error covariance on the interval $t_{k-1} \leq t \leq t_k$, we numerically integrate

$$\begin{aligned} \dot{\Phi}(t, t_{k-1}) &= \mathbf{F}(\hat{\mathbf{x}}(t), t)\Phi(t, t_{k-1}) \\ \dot{\mathbf{Q}}(t) &= \mathbf{F}(\hat{\mathbf{x}}(t), t)\mathbf{Q}(t) + \mathbf{Q}(t)\mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{M}(t)\mathbf{Q}_{spec}(t)\mathbf{M}^T(t) \\ \dot{\hat{\mathbf{x}}}(t) &= \mathbf{f}(\hat{\mathbf{x}}(t), t) \end{aligned}$$


from $t = t_{k-1}$ to $t = t_k$ with initial conditions

$$\begin{aligned} \Phi(t_{k-1}, t_{k-1}) &= \mathbf{I}_{n \times n} \\ \mathbf{Q}(t_{k-1}) &= \mathbf{0}_{n \times n} \\ \hat{\mathbf{x}}(t_{k-1}) &= \hat{\mathbf{x}}_{k-1}^+ \end{aligned} \quad (10)$$

to obtain $\mathbf{Q}(t_k)$ and $\Phi(t_k, t_{k-1})$. After integration, the state estimation error covariance is mapped forward via

$$\begin{aligned} \mathbf{P}_k^- &= \Phi_k \mathbf{P}_{k-1}^+ \Phi_k^T + \mathbf{Q}_k \\ \hat{\mathbf{x}}_k^- &= \hat{\mathbf{x}}(t_k) \end{aligned} \quad (11)$$

where $\mathbf{Q}_k = \mathbf{Q}(t_k)$ and $\Phi_k = \Phi(t_k, t_{k-1})$.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 53 of 73

2.3 Update

Taking the expected value of both sides of Eq. (2) yields

$$\hat{y}_k = E\{y_k\} = E\{h_k(x_k)\} + E\{v_k\}.$$

Using a Taylor series expansion of $h_k(x_k)$ about the *a priori* state estimate, \hat{x}_k^- , and neglecting higher-order terms it is seen that

$$\hat{y}_k = E\{h_k(\hat{x}_k^-) + H_k(\hat{x}_k^-)(x_k - \hat{x}_k^-) + E\{v_k\}\}. \quad (12)$$

where $H_k(\hat{x}_k^-)$ is the measurement sensitivity matrix, defined to be

$$H_k(\hat{x}_k^-) = \left. \frac{\partial h_k(x_k)}{\partial x_k} \right|_{x_k = \hat{x}_k^-}.$$

Eq. (12) can be written as

$$\hat{y}_k = h_k(\hat{x}_k^-) + H_k(\hat{x}_k^-)E\{(x_k - \hat{x}_k^-)\} + E\{v_k\}.$$

Since the EKF is an unbiased estimator ($E\{(x_k - \hat{x}_k^-)\} = 0$) and the measurement noise is assumed to be zero-mean ($E\{v_k\} = 0$), the estimated measurement is found to be

$$\hat{y}_k = h_k(\hat{x}_k^-).$$

Note that both the measurement and the estimated measurement are evaluated just prior to the measurement update. Defining the measurement residual, r_k , and the Kalman gain matrix, K_k , to be

$$r_k = y_k - \hat{y}_k \quad (13)$$

$$K_k = P_k^- H_k^T(\hat{x}_k^-) W_k^{-1}, \quad (14)$$

where W_k is the measurement residual covariance matrix,


$$W_k = H_k(\hat{x}_k^-) P_k^- H_k^T(\hat{x}_k^-) + R_k, \quad (15)$$

the state estimate and the state estimation error covariance update are then calculated as

$$\hat{x}_k^+ = \hat{x}_k^- + K_k r_k \quad (16)$$

$$P_k^+ = [I_{n \times n} - K_k H_k(\hat{x}_k^-)] P_k^- [I_{n \times n} - K_k H_k(\hat{x}_k^-)]^T + K_k R_k K_k^T. \quad (17)$$

In the EKF development, we must specify the dynamics model $f(x(t), t)$, the measurement model, $h_k(x_k)$, the process noise matrix, Q_{spec} , the process noise mapping matrix, $M(t)$, and the measurement noise covariance R_k . To execute the EKF we need to specify the initial conditions $\hat{x}(t_0) = E\{x(t_0)\}$ and $P_0 = E\{(x(t_0) - \hat{x}(t_0))(x(t_0) - \hat{x}(t_0))^T\}$. As the measurements become available at t_k , we can propagate the state estimate and state estimation error covariance, \hat{x}_{k-1}^+ and P_{k-1}^+ , respectively, from t_{k-1} to t_k to obtain \hat{x}_k^- and P_k^- , and use the measurement information to update the state estimate and state estimation error covariance to obtain \hat{x}_k^+ and P_k^+ . This process continues as the measurements continue to become available. Overall, the EKF architecture is algorithmically straightforward. The challenge is to determine a viable set of states and associated models for the dynamics and sensors so that the EKF can be implemented in real-time. In this work, we also strive to create a *structured* architecture that permits the EKF to be readily reconfigured for different sensors and scenarios.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 54 of 73

3 Mathematical Model of the Flight Dynamics

3.1 Dynamical Model in Continuous-time

The dynamics of the spacecraft during EDL occur naturally in continuous-time. However, since the EKF executes on a digital computer, the dynamics modeled in the estimation algorithm occur naturally in discrete-time. Also, the IMU and external sensors provide measurements at discrete times. The strategy here is to present the dynamics of the spacecraft during EDL in continuous-time and then with appropriate assumptions create a discrete-time model equivalent. One important aspect of the approach followed here is to utilize the IMU to provide a measurement of all the non-gravitational accelerations (thrust, aerodynamics, etc.) acting on the spacecraft rather than attempting to model their associated dynamics explicitly.

In the mEKF we navigate the IMU (not the center of gravity). The total acceleration is comprised of accelerations due to gravity and accelerations due to non-conservative forces, such as thrust and aerodynamics. The acceleration due to gravity is a function of the position of the center-of-gravity of the vehicle, where

$$\mathbf{r}_{cg}^i = \mathbf{r}_{imu}^i(t) + \mathbf{T}_b^i(\mathbf{q}_t^b) \mathbf{r}_{cg/imu}^b.$$

The IMU is not generally co-located with the CG of the vehicle; instead, it is offset at an offset position from the CG, denoted by $\mathbf{r}_{cg/imu}^b$, where the superscript b denotes body reference frame. The transformation matrix from body reference frame to inertial reference frame, denoted by $\mathbf{T}_b^i(\mathbf{q}_t^b)$, is given by

$$\mathbf{T}_b^i(\mathbf{q}_t^b) = \left[(q^2 + \mathbf{q}^T \mathbf{q}) \mathbf{I}_{3 \times 3} - 2q[\mathbf{q} \times] + 2[\mathbf{q} \times]^2 \right]^T \in \mathbb{R}^{3 \times 3}, \quad (18)$$

where

$$\mathbf{q}_t^b := \begin{bmatrix} \mathbf{q} \\ q \end{bmatrix} \in \mathbb{R}^{4 \times 1}$$

is the attitude quaternion representing the inertial reference frame relative to the case reference frame. The continuous equations of motion are given by

$$\dot{\mathbf{r}}_{imu}^i(t) = \mathbf{v}_{imu}^i(t) \quad (19)$$

$$\dot{\mathbf{v}}_{imu}^i(t) = \mathbf{a}_g^i(\mathbf{r}_{imu}^i(t) + \mathbf{T}_b^i(\mathbf{q}_t^b) \mathbf{r}_{cg/imu}^b) + \mathbf{T}_b^i(\mathbf{q}_t^b) \mathbf{T}_c^b \mathbf{a}_{ng}^c(t) \quad (20)$$

$$\dot{\mathbf{q}}_t^b = \frac{1}{2} \bar{\omega}_{b/i}^b(t) \otimes \mathbf{q}_t^b(t) \quad (21)$$

where \mathbf{T}_c^b is a known constant matrix and $\bar{\omega}$ is the pure quaternion given by

$$\bar{\omega}_{b/i}^b = \begin{bmatrix} \omega_{b/i}^b \\ 0 \end{bmatrix}.$$


The non-conservative forces (accelerations denoted by $\mathbf{a}_{ng}^c(t)$ in Eq. (20)) are sensed by the IMU. The IMU then becomes an instrumental element of the state propagation since it essentially replaces a mathematical model of the non-conservative forces.

Note that the quaternion of rotation is subject to a unity norm constraint or

$$\bar{q} = |\mathbf{q}| = \sqrt{q^2 + \mathbf{q}^T \mathbf{q}} = 1.$$

Quaternion multiplication (i.e. the composition of rotations) for any two arbitrary quaternions, \bar{p} and \bar{r} is given by

$$\bar{p} \otimes \bar{r} = \begin{bmatrix} \mathbf{p} \\ p \end{bmatrix} \otimes \begin{bmatrix} \mathbf{r} \\ r \end{bmatrix} = \begin{bmatrix} \mathbf{pr} + p\mathbf{r} - \mathbf{p} \times \mathbf{r} \\ pr - \mathbf{p}^T \mathbf{r} \end{bmatrix}, \quad (22)$$

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 55 of 73

where the symbol \otimes is exclusively used to denote quaternion multiplication. In the special case that the quaternion is used to represent a small rotation, denoted by $\delta\alpha$, the small angle quaternion, denoted by $\delta\bar{q}$, is given by

$$\delta\bar{q} = \begin{bmatrix} \delta q \\ \delta q \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\delta\alpha \\ 1 \end{bmatrix}. \quad (23)$$

Substituting Eq. (23) into Eq. (18) yields (to first order in $\delta\alpha$), the rotation matrix, $\delta\mathbf{T}$, representing the small angle rotation

$$\delta\mathbf{T} = \mathbf{I}_{3 \times 3} - [\delta\alpha \times], \quad (24)$$

where

$$[\delta\alpha \times] = \begin{bmatrix} 0 & -\delta a_3 & \delta a_2 \\ \delta a_3 & 0 & -\delta a_1 \\ -\delta a_2 & \delta a_1 & 0 \end{bmatrix}.$$

Therefore, given a set of small angles which may represent a deviation in attitude, the corresponding quaternion is given via Eq. (23) and the corresponding rotation matrix is given via Eq. (24). This is important because the attitude representation using quaternions increases the complexity of the estimation process since the attitude update occurs through multiplication rather than through addition. The mEKF will utilize a multiplicative update for the attitude estimation based on small angle approximations.

3.2 Inertial Measurement Units

The strapdown IMU accelerometers and gyros produce measurements corrupted by random and systematic errors. The accelerometer directly measures the non-gravitational accelerations acting on the spacecraft. We assume that the accelerometer package outputs a measure of the spacecraft change in velocity due to non-gravitational accelerations (expressed in the IMU case frame) over the time interval t_{k-1} to t_k denoted by $\Delta v_{m,k}^c$. We model the computation of the true $\Delta v_{true,k}^c$ via

$$\Delta v_{true,k}^c = \int_{t_{k-1}}^{t_k} \mathbf{a}^c(\tau) d\tau. \quad (25)$$

The actual measurement of acceleration is corrupted by systematic and random errors. To model the errors we assume a model of the form

$$\mathbf{a}_m^c(t) = (\mathbf{I}_{3 \times 3} + \mathbf{\Gamma}_a)(\mathbf{I}_{3 \times 3} + \mathbf{S}_a)(\mathbf{a}^c(t) + \boldsymbol{\beta}_a + \boldsymbol{\xi}_t), \quad (26)$$

where


$$\mathbf{\Gamma}_a := \begin{bmatrix} 0 & \gamma_{a_{xz}} & -\gamma_{a_{xy}} \\ -\gamma_{a_{yz}} & 0 & \gamma_{a_{yz}} \\ \gamma_{a_{zy}} & -\gamma_{a_{zx}} & 0 \end{bmatrix}, \quad \mathbf{S}_a := \begin{bmatrix} s_{a_x} & 0 & 0 \\ 0 & s_{a_y} & 0 \\ 0 & 0 & s_{a_z} \end{bmatrix},$$

and $\boldsymbol{\gamma}_a = (\gamma_{a_{yz}}, \gamma_{a_{zy}}, \gamma_{a_{xz}}, \gamma_{a_{zx}}, \gamma_{a_{xy}}, \gamma_{a_{yx}})^T$ are nonorthogonality and axes misalignment errors, $\boldsymbol{\beta}_a \in \mathbb{R}^3$ is the bias in the accelerometer, $\mathbf{s}_a = (s_{a_x}, s_{a_y}, s_{a_z})^T$ are scale factor errors, and $\boldsymbol{\xi}_t \in \mathbb{R}^3$ is a zero-mean, white noise process with $E[\boldsymbol{\xi}_t] = 0$ and

$$E[\boldsymbol{\xi}_t \boldsymbol{\xi}_\tau^T] = \boldsymbol{\Xi} \delta(t - \tau) \quad \text{where} \quad \delta(t - \tau) = 0 \text{ when } t \neq \tau \text{ and } \delta(t - \tau) = 1 \text{ when } t = \tau.$$

The nonorthogonality and axes misalignment errors, scale factor errors, and bias parameters are all modeled as zero-mean random constants. We characterize the random constants with $E[\boldsymbol{\gamma}_a] = 0$, $E[\mathbf{s}_a] = 0$, $E[\boldsymbol{\beta}_a] = 0$ and the given matrices $E[\boldsymbol{\gamma}_a \boldsymbol{\gamma}_a^T]$, $E[\mathbf{s}_a \mathbf{s}_a^T]$, and $E[\boldsymbol{\beta}_a \boldsymbol{\beta}_a^T]$. If we assume that the various errors are small, then to first-order

$$(\mathbf{I}_{3 \times 3} + \mathbf{\Gamma}_a)(\mathbf{I}_{3 \times 3} + \mathbf{S}_a) \approx \mathbf{I}_{3 \times 3} + \mathbf{\Gamma}_a + \mathbf{S}_a.$$

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 56 of 73

Defining

$$\Delta_a := \Gamma_a + S_a \quad (27)$$

yields the accelerometer measurement model

$$\mathbf{a}_m^c(t) = (\mathbf{I}_{3 \times 3} + \Delta_a) (\mathbf{a}^c(t) + \boldsymbol{\beta}_a + \boldsymbol{\xi}_t), \quad (28)$$

Typically, the IMU error parameters (such as the bias covariance and noise spectral density) associated with the continuous-time acceleration error model in Eq. (28) are provided as part of the hardware specifications. The challenge here is to develop an error model that reflects the output of the IMU given in terms of integrated acceleration, understanding that there may be internal computations occurring within the IMU unit and that the IMU outputs are available only at discrete times. To that end, model the measured $\Delta \mathbf{v}_{m,k}^c$ at time t_k via

$$\Delta \mathbf{v}_{m,k}^c = \int_{t_{k-1}}^{t_k} \mathbf{a}_m^c(\tau) d\tau \quad (29)$$

where $\mathbf{a}_m^c(t)$ is given by Eq. (28). The associated accelerometer measurement model, $\Delta \mathbf{v}_{m,k}^c$, is represented by

$$\Delta \mathbf{v}_{m,k}^c = (\mathbf{I} + \Delta_a)(\Delta \mathbf{v}_{true,k}^c + \mathbf{b}_a + \boldsymbol{\xi}_k). \quad (30)$$

where $\Delta_a(\gamma_a, s_a)$ are nonorthogonality and axes misalignment errors and scale factor errors given as above in Eq. (28), $\mathbf{b}_a \in \mathbb{R}^3$ is the bias in the accelerometer and is a function of the integration time, and $\boldsymbol{\xi}_k \in \mathbb{R}^3$ is a zero-mean, white noise sequence. From Eq. (30), compute $\Delta \mathbf{v}_{m,k}^c$ as

$$\Delta \mathbf{v}_{m,k}^c = \Delta \mathbf{v}_{true,k}^c + \mathbf{D}(\Delta \mathbf{v}_{m,k}^c) s_a + \mathbf{N}(\Delta \mathbf{v}_{m,k}^c) \gamma_a + \mathbf{b}_a + \boldsymbol{\xi}_k \quad (31)$$

where we have utilized the approximation $(\mathbf{I} + \Delta_a)^{-1} \simeq \mathbf{I} - \Delta_a$ and

$$\mathbf{D}(\Delta \mathbf{v}_{m,k}^c) := \begin{bmatrix} \Delta v_x & 0 & 0 \\ 0 & \Delta v_y & 0 \\ 0 & 0 & \Delta v_z \end{bmatrix}, \quad \mathbf{N}(\Delta \mathbf{v}_{m,k}^c) := \begin{bmatrix} -\Delta v_z & \Delta v_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta v_z & -\Delta v_x & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Delta v_y & \Delta v_x \end{bmatrix},$$

where $\Delta \mathbf{v}_{m,k}^c = (\Delta v_x, \Delta v_y, \Delta v_z)^T$. In the current version of the mEKF, we estimate only the IMU accelerometer bias, \mathbf{b}_a , and treat s_a , γ_a , and $\boldsymbol{\xi}_k$ as contributing to the process noise of the dynamics. Hence, we have the model

$$\Delta \mathbf{v}_{m,k}^c = \Delta \mathbf{v}_{true,k}^c + \mathbf{b}_a + \mathbf{w}_{\Delta v,k} \quad (32)$$

where $\mathbf{w}_{\Delta v,k} = \mathbf{D}(\Delta \mathbf{v}_{m,k}^c) s_a + \mathbf{N}(\Delta \mathbf{v}_{m,k}^c) \gamma_a + \boldsymbol{\xi}_k$ and we assume that $E[\mathbf{w}_{\Delta v,k} \mathbf{w}_{\Delta v,j}^T] = \mathbf{W}_{\Delta v} \delta_{kj}$ where $\delta_{kj} = 0$ when $k \neq j$ and $\delta_{kj} = 1$ when $k = j$ and we choose $\mathbf{W}_{\Delta v}$ appropriately.

The situation with the gyros is similar to that of the accelerometers. The gyro measures the angular velocity of the spacecraft; however, the gyro package outputs a measure of the spacecraft attitude change over the time interval t_{k-1} to t_k denoted by $\Delta \boldsymbol{\theta}_{m,k}^c$. The measurement of the angular velocity vector is corrupted by random biases, errors due to scale factor uncertainties, errors due to nonorthogonality and axes misalignments, and random noise. The gyro error model can be formulated as


$$\Delta \boldsymbol{\theta}_{m,k}^c = \int_{t_{k-1}}^{t_k} \boldsymbol{\omega}^c(\tau) d\tau \quad (33)$$

and the corresponding measurement model is

$$\Delta \boldsymbol{\theta}_{m,k}^c = \int_{t_{k-1}}^{t_k} \boldsymbol{\omega}_m^c(\tau) d\tau$$

from which it follows that

$$\Delta \boldsymbol{\theta}_{m,k}^c = (\mathbf{I}_{3 \times 3} + \Gamma_g)(\mathbf{I}_{3 \times 3} + \mathbf{S}_g)(\Delta \boldsymbol{\theta}_{true,k}^c + \mathbf{b}_g + \boldsymbol{\eta}_k), \quad (34)$$

	<h1 style="text-align: center;">NASA Engineering and Safety Center</h1> <h2 style="text-align: center;">Technical Assessment Report</h2>	Document #: NESC-RP-09-00530	Version: 1.0
Title: <h3 style="text-align: center;">Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis</h3>			Page #: 57 of 73

where \mathbf{b}_g is the gyro bias, \mathbf{S}_g is the gyro scale factor matrix, $\mathbf{\Gamma}_g$ is the gyro nonorthogonality and axes misalignment matrix, $\boldsymbol{\eta}_k$ is a white sequence, and where

$$\mathbf{S}_g := \begin{bmatrix} s_{g_x} & 0 & 0 \\ 0 & s_{g_y} & 0 \\ 0 & 0 & s_{g_z} \end{bmatrix}, \quad \mathbf{\Gamma}_g := \begin{bmatrix} 0 & \gamma_{g_{xz}} & -\gamma_{g_{xy}} \\ -\gamma_{g_{yz}} & 0 & \gamma_{g_{yx}} \\ \gamma_{g_{xy}} & -\gamma_{g_{xz}} & 0 \end{bmatrix}, \quad (35)$$

and $\boldsymbol{\gamma}_g = (\gamma_{g_{xz}}, \gamma_{g_{yz}}, \gamma_{g_{xz}}, \gamma_{g_{yz}}, \gamma_{g_{xz}}, \gamma_{g_{yz}})^T$ are nonorthogonality and axes misalignment errors, $\mathbf{b}_g \in \mathbb{R}^3$ is the bias in the gyro, $\mathbf{s}_g = (s_{g_x}, s_{g_y}, s_{g_z})^T$ are scale factor errors, and $\boldsymbol{\eta}_k \in \mathbb{R}^3$ is a zero-mean, white noise sequence. The nonorthogonality and axes misalignment errors, scale factor errors, and bias parameters are all modeled as zero-mean random constants. To first-order, we have

$$(\mathbf{I}_{3 \times 3} + \mathbf{S}_g)(\mathbf{I}_{3 \times 3} + \mathbf{\Gamma}_g) \approx \mathbf{I}_{3 \times 3} + \mathbf{S}_g + \mathbf{\Gamma}_g.$$

Hence, Eq. (34) can be written in the form

$$\Delta \theta_{m,k}^c = (\mathbf{I}_{3 \times 3} + \Delta_g)(\Delta \theta_{true,k}^c + \mathbf{b}_g + \boldsymbol{\eta}_k), \quad (36)$$

where $\Delta_g := \mathbf{S}_g + \mathbf{\Gamma}_g$. From Eq. (36), compute

$$\Delta \theta_{m,k}^c = \Delta \theta_{true,k}^c + \mathbf{D}(\Delta \theta_{m,k}^c) \mathbf{s}_g + \mathbf{N}(\Delta \theta_{m,k}^c) \boldsymbol{\gamma}_g + \mathbf{b}_g + \boldsymbol{\eta}_k, \quad (37)$$

where we have utilized the approximation $(\mathbf{I} + \Delta_g)^{-1} \simeq \mathbf{I} - \Delta_g$ and

$$\mathbf{D}(\Delta \theta_{m,k}^c) := \begin{bmatrix} \Delta \theta_z & 0 & 0 \\ 0 & \Delta \theta_y & 0 \\ 0 & 0 & \Delta \theta_x \end{bmatrix}, \quad \mathbf{N}(\Delta \theta_{m,k}^c) := \begin{bmatrix} -\Delta \theta_z & \Delta \theta_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta \theta_z & -\Delta \theta_x & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Delta \theta_y & \Delta \theta_x \end{bmatrix},$$

where $\Delta \theta_{m,k}^c = (\Delta \theta_x, \Delta \theta_y, \Delta \theta_z)^T$. In the current version of the mEKF, we estimate only the IMU gyro bias, \mathbf{b}_g , and treat \mathbf{s}_g , $\boldsymbol{\gamma}_g$, and $\boldsymbol{\eta}_k$ as contributing to the process noise of the dynamics. Hence, we have the model

$$\Delta \theta_{m,k}^c = \Delta \theta_{true,k}^c + \mathbf{b}_g + \mathbf{w}_{\Delta \theta,k}, \quad (38)$$

where $\mathbf{w}_{\Delta \theta,k} = \mathbf{D}(\Delta \theta_{m,k}^c) \mathbf{s}_g + \mathbf{N}(\Delta \theta_{m,k}^c) \boldsymbol{\gamma}_g + \boldsymbol{\eta}_k$, and we assume that $E[\mathbf{w}_{\Delta \theta,k} \mathbf{w}_{\Delta \theta,j}^T] = \mathbf{W}_{\Delta \theta} \delta_{kj}$ where $\delta_{kj} = 0$ when $k \neq j$ and $\delta_{kj} = 1$ when $k = j$ and we choose $\mathbf{W}_{\Delta \theta}$ appropriately.


3.2.1 Accumulated Integrated Accelerometers and Gyros

Consider the scenario in which an IMU is operating at a frequency that is a multiple of that of other subsystems. In this case, it is possible that the IMU is capable of streaming multiple measurements of the integrated non-gravitational acceleration and angular velocity over a single system time-step. Consider the accumulated measured integrated non-gravitational acceleration as

$$\Sigma \mathbf{v}_{m,k}^c := \sum_{\ell=1}^n \Delta \mathbf{v}_{m,\ell}^c$$

where it has been assumed that there are n subdivisions of the interval $\{t_{k-1}, t_k\}$ with a single subinterval defined by $\{t_{\ell-1}, t_\ell\}$ and where $\Delta \mathbf{v}_{m,\ell}^c$ is given by Eq. (32). Let

$$\Sigma \mathbf{v}_k^c := \sum_{\ell=1}^n \Delta \mathbf{v}_\ell^c, \quad \mathbf{b}_{\Sigma \Delta v} := \sum_{\ell=1}^n \mathbf{b}_{\Delta v}, \quad \text{and} \quad \mathbf{w}_{\Sigma \Delta v,k} := \sum_{\ell=1}^n \mathbf{w}_{\Delta v,\ell}$$

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 58 of 73

such that

$$\Sigma \mathbf{v}_{m,k}^c = \Sigma \mathbf{v}_k^c + \mathbf{b}_{\Sigma \Delta V} + \mathbf{w}_{\Sigma \Delta V,k}.$$

Similarly, consider the accumulated measured integrated angular velocity as

$$\Sigma \theta_{m,k}^c := \sum_{\ell=1}^n \Delta \theta_{m,\ell}^c$$

where it has been assumed that there are n subdivisions of the interval $\{t_{k-1} \ t_k\}$ with a single subinterval defined by $\{t_{\ell-1} \ t_\ell\}$ and where $\Delta \theta_{m,\ell}^c$ is given by Eq. (38). Let

$$\Sigma \theta_k^c := \sum_{\ell=1}^n \Delta \theta_\ell^c, \quad \mathbf{b}_{\Sigma \theta} := \sum_{\ell=1}^n \mathbf{b}_{\Delta \theta}, \quad \text{and} \quad \mathbf{w}_{\Sigma \theta,k} := \sum_{\ell=1}^n \mathbf{w}_{\Delta \theta,\ell}$$

such that

$$\Sigma \theta_{m,k}^c = \Sigma \theta_k^c + \mathbf{b}_{\Sigma \theta} + \mathbf{w}_{\Sigma \theta,k}.$$

3.3 Dynamical Model in Discrete-time

Define the quaternion $\Delta q_i^b(t)$ representing the rotation from the *a priori* attitude, that is the attitude at time t_{k-1} denoted by $\bar{q}_{i,k-1}^b$, as

$$\Delta q_i^b(t) := \bar{q}_i^b(t) \otimes \bar{q}_{i,k-1}^{b^{-1}}$$

Let $\theta^b(t)$ be the rotation vector parameterization of $\Delta q_i^b(t)$. Assume a small time step in the state propagation with the IMU from t_{k-1} to t_k . Linearizing Bortz's equation around small angles and assuming $\omega_{b/i}^b(t) = \omega_{b/i,k}^b$ yields

$$\dot{\theta}^b(t) = \omega_{b/i,k}^b + \frac{1}{2} \theta^b(t) \times \omega_{b/i,k}^b \quad t_{k-1} \leq t < t_k$$

which has the solution

$$\theta^b(t) = \omega_{b/i,k}^b (t - t_{k-1}) \quad t_{k-1} \leq t < t_k. \quad (39)$$

Letting $t \rightarrow t_k$ and defining $\Delta t_k := t_k - t_{k-1}$ yields

$$\theta_k^b = \omega_{b/i,k}^b \Delta t_k = \Sigma \theta_k^b,$$


where we assume the IMU possesses accumulated integrating gyros. Therefore, the discretized quaternion propagation is given by

$$\bar{q}_{i,k}^b = \bar{q}_i^b(\Sigma \theta_k^b) \otimes \bar{q}_{i,k-1}^b \quad (40)$$

where

$$\bar{q}_i^b(\Sigma \theta_k^b) = \begin{bmatrix} \sin\left(\frac{1}{2} \|\Sigma \theta_k^b\|\right) \Sigma \theta_k^b / \|\Sigma \theta_k^b\| \\ \cos\left(\frac{1}{2} \|\Sigma \theta_k^b\|\right) \end{bmatrix}.$$

Note that the IMU output $\Sigma \theta_k^c$ is in the IMU case frame, so when utilizing Eq. (40) we must first transform the output using the constant (and known) \mathbf{T}_c^b as $\Sigma \theta_k^b = \mathbf{T}_c^b \Sigma \theta_k^c$. As with the attitude discretization, assume a small time step in the position and velocity propagation with the IMU from t_{k-1} to t_k . Over this

	<h1 style="text-align: center;">NASA Engineering and Safety Center</h1> <h2 style="text-align: center;">Technical Assessment Report</h2>	Document #: NESC-RP-09-00530	Version: 1.0
Title: <h3 style="text-align: center;">Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis</h3>			Page #: 59 of 73

small time step $\Delta t_k := t_k - t_{k-1}$, assume that the non-gravitational acceleration $a_{ng}^c(t) = a_{ng,k}^c$ and the gravity $a_g^i(t) = g_{k-1}^i := g(r_{imu,k-1}^i + T_{b,k-1}^i r_{cg/imu}^b)$, where $T_{b,k-1}^i := T_b^i(\bar{q}_{i,k-1}^b)$. Since we assume small angles over Δt_k , we can use Eq. (24) to obtain

$$T_b^i(\bar{q}_i^b) = \left(I_{3 \times 3} - [\theta^b \times] \right) T_{b,i,k-1}^i \quad t_{k-1} \leq t < t_k. \quad (41)$$

With $T_b^i(\bar{q}_i^b) = T_{b,i,k-1}^i$ and $\omega_{b/i}^b(t) = \omega_{b/i,k}^b$, write Eq. (20) as

$$\dot{r}_{imu}^i(t) = g_{k-1}^i + T_{b,k-1}^i T_c^b a_{ng,k}^c - T_{b,k-1}^i [T_c^b a_{ng,k}^c \times] \omega_{b/i,k}^b (t - t_{k-1}). \quad (42)$$

In Eq. (42), the constant gravity is approximated by $g_{k-1}^i \approx g^i(r_{imu,k-1}^i)$. Integrating Eq. (42) twice and once, respectively, yields the position and velocity evolution over the interval $t_{k-1} \leq t < t_k$ as

$$\begin{aligned} r_{imu}^i(t) &= r_{imu,k-1}^i + v_{imu,k-1}^i(t - t_{k-1}) + \frac{1}{2} g_{k-1}^i (t - t_{k-1})^2 + \frac{1}{2} T_{b,k-1}^i T_c^b a_{ng,k}^c (t - t_{k-1})^2 \\ &\quad - \frac{1}{6} T_{b,k-1}^i [T_c^b a_{ng,k}^c \times] \omega_{b/i,k}^b (t - t_{k-1})^3 \\ v_{imu}^i(t) &= v_{imu,k-1}^i + g_{k-1}^i (t - t_{k-1}) + T_{b,k-1}^i T_c^b a_{ng,k}^c (t - t_{k-1}) - \frac{1}{2} T_{b,k-1}^i [T_c^b a_{ng,k}^c \times] \omega_{b/i,k}^b (t - t_{k-1})^2 \end{aligned}$$

Letting $t \rightarrow t_k$, we have

$$\begin{aligned} r_{imu,k}^i &= r_{imu,k-1}^i + v_{imu,k-1}^i \Delta t_k + \frac{1}{2} g_{k-1}^i \Delta t_k^2 + \frac{1}{2} T_{b,k-1}^i T_c^b a_{ng,k}^c \Delta t_k^2 - \frac{1}{6} T_{b,k-1}^i [T_c^b a_{ng,k}^c \times] \omega_{b/i,k}^b \Delta t_k^3 \\ v_{imu,k}^i &= v_{imu,k-1}^i + g_{k-1}^i \Delta t_k + T_{b,k-1}^i T_c^b a_{ng,k}^c \Delta t_k - \frac{1}{2} T_{b,k-1}^i [T_c^b a_{ng,k}^c \times] \omega_{b/i,k}^b \Delta t_k^2 \end{aligned}$$

Assuming that the IMU possesses accumulating integrating gyros and accelerometers, we rewrite the position and velocity propagation in their final form as

$$r_{imu,k}^i = r_{imu,k-1}^i + v_{imu,k-1}^i \Delta t_k + \frac{1}{2} g_{k-1}^i \Delta t_k^2 + \frac{1}{2} T_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{3} [T_c^b \Sigma \theta_k^c \times] \right) T_c^b \Sigma v_k^c \Delta t_k \quad (43a)$$

$$v_{imu,k}^i = v_{imu,k-1}^i + g_{k-1}^i \Delta t_k + T_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{2} [T_c^b \Sigma \theta_k^c \times] \right) T_c^b \Sigma v_k^c \quad (43b)$$

The discretized mathematical model describing the evolution of the position, velocity, and attitude is given by Eq. (40) and Eqs. (43). The estimated states in the EKF are propagated accordingly via

$$\hat{r}_{imu,k}^i = \hat{r}_{imu,k-1}^i + \hat{v}_{imu,k-1}^i \Delta t_k + \frac{1}{2} \hat{g}_{k-1}^i \Delta t_k^2 + \frac{1}{2} \hat{T}_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{3} [T_c^b \Sigma \hat{\theta}_k^c \times] \right) T_c^b \Sigma \hat{v}_k^c \Delta t_k \quad (44a)$$

$$\hat{v}_{imu,k}^i = \hat{v}_{imu,k-1}^i + \hat{g}_{k-1}^i \Delta t_k + \hat{T}_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{2} [T_c^b \Sigma \hat{\theta}_k^c \times] \right) T_c^b \Sigma \hat{v}_k^c \quad (44b)$$

$$\hat{q}_{i,k}^c = \bar{q}(\Sigma \hat{\theta}_k) \otimes \hat{q}_{i,k-1}^c \quad (44c)$$


where $\hat{g}_{k-1}^i := g(\hat{r}_{imu,k-1}^i)$ and $\hat{T}_{b,k-1}^i := T_b^i(\hat{q}_{i,k-1}^b)$. Note that the accumulated integrated output of the IMU, denoted by $\Sigma \theta_k^c$ and Σv_k^c , is corrected with the estimated IMU gyro and accelerometer biases to obtain the values $\Sigma \hat{\theta}_k^c$ and $\Sigma \hat{v}_k^c$ for use in the EKF propagation.

3.4 State Estimation Errors

3.4.1 State Estimation Error Propagation

Define the multiplicative attitude estimation error at any time t_k as

$$\delta q_{i,k}^b := q_{i,k}^b \otimes \hat{q}_{i,k}^{b^{-1}}.$$

	<h1 style="text-align: center;">NASA Engineering and Safety Center</h1> <h2 style="text-align: center;">Technical Assessment Report</h2>	Document #: NESC-RP-09-00530	Version: 1.0
Title: <h3 style="text-align: center;">Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis</h3>			Page #: 60 of 73

For small angles, the attitude error is approximately twice the vector part of the quaternion $\delta \hat{q}_{i,k}^b$, so

$$e_{\theta,k} := 2\delta \hat{q}_{i,k}^b.$$

Also define

$$e_{\Sigma\theta,k} := \Sigma\hat{\theta}_k^b - \Sigma\hat{\theta}_k^b \quad \text{and} \quad e_{b_{\Sigma\theta},k} := b_{\Sigma\theta} - \hat{b}_{\Sigma\theta,k}.$$

Using $\Sigma\hat{\theta}_k^b$ to compute

$$q_{k,i}^b(\Sigma\hat{\theta}_k^b) = \begin{bmatrix} \sin\left(\frac{1}{2}\|\Sigma\hat{\theta}_k^b\|\right) \frac{\Sigma\hat{\theta}_k^b}{\|\Sigma\hat{\theta}_k^b\|} \\ \cos\left(\frac{1}{2}\|\Sigma\hat{\theta}_k^b\|\right) \end{bmatrix},$$

we denote the associated transformation matrix by $T(\Sigma\hat{\theta}_k^b)$. The attitude estimation error propagation (from t_{k-1} after the previous state update in the EKF to t_k prior to the next state update) is given by

$$e_{\theta,k}^- = T(\Sigma\hat{\theta}_k^b) e_{\theta,k-1}^+ - T_c^b e_{b_{\Sigma\theta},k-1}^+ - T_c^b w_{\Sigma\theta,k}. \quad (45)$$

Define the position and velocity estimation errors as

$$e_{r,k} := r_{imu,k}^i - \hat{r}_{imu,k}^i \quad \text{and} \quad e_{v,k} := v_{imu,k}^i - \hat{v}_{imu,k}^i$$

Also define

$$e_{\Sigma v,k} := \Sigma v_k^b - \Sigma \hat{v}_k^b \quad \text{and} \quad e_{b_{\Sigma v},k} := b_{\Sigma v} - \hat{b}_{\Sigma v,k},$$

where $\Sigma v_k^b = T_c^b \Sigma v_k^c$ is transformed from the IMU case frame. The position and velocity estimation errors propagate according to

$$\begin{aligned} e_{r,k}^- &= e_{r,k-1}^+ + e_{v,k-1}^+ \Delta t_k - \frac{1}{2} \left(\hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] + \frac{1}{3} \hat{T}_{b,k-1}^i [(\Sigma \hat{\theta}_k^b \times \Sigma \hat{v}_k^b) \times] \right) \Delta t_k e_{\theta,k-1}^+ \\ &\quad - \hat{R}_a e_{b_{\Sigma v},k-1}^+ - \hat{R}_g e_{b_{\Sigma\theta},k-1}^+ + \hat{R}_a w_{\Sigma\Delta V,k} + \hat{R}_g w_{\Sigma\theta,k} \\ e_{v,k}^- &= e_{v,k-1}^+ - \left(\hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] + \frac{1}{2} \hat{T}_{b,k-1}^i [(\Sigma \hat{\theta}_k^b \times \Sigma \hat{v}_k^b) \times] \right) e_{\theta,k-1}^+ \\ &\quad - \hat{V}_a e_{b_{\Sigma v},k-1}^+ - \hat{V}_g e_{b_{\Sigma\theta},k-1}^+ + \hat{V}_a w_{\Sigma\Delta V,k} + \hat{V}_g w_{\Sigma\theta,k} \end{aligned} \quad (46)$$

where


$$\begin{aligned} \hat{R}_a &= \frac{1}{2} \hat{T}_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{3} [\Sigma \hat{\theta}_k^b \times] \right) T_c^b \Delta t_k, \quad \hat{R}_g = \frac{1}{6} \hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] T_c^b \Delta t_k, \\ \hat{V}_a &= \hat{T}_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{2} [\Sigma \hat{\theta}_k^b \times] \right) T_c^b, \quad \text{and} \quad \hat{V}_g = \frac{1}{2} \hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] T_c^b. \end{aligned}$$

The IMU and sensor biases are modeled as random constants. Define the estimation errors in the velocimeter, attitude camera, and altimeter biases, respectively, as

$$\begin{aligned} e_{b_{vel},k}^- &:= b_{vel} - \hat{b}_{vel,k}^- \quad \text{and} \quad e_{b_{vel},k-1}^+ := b_{vel} - \hat{b}_{vel,k-1}^+ \\ e_{b_{ac},k}^- &:= b_{ac} - \hat{b}_{ac,k}^- \quad \text{and} \quad e_{b_{ac},k-1}^+ := b_{ac} - \hat{b}_{ac,k-1}^+ \\ e_{b_{alt},k}^- &:= b_{alt} - \hat{b}_{alt,k}^- \quad \text{and} \quad e_{b_{alt},k-1}^+ := b_{alt} - \hat{b}_{alt,k-1}^+, \end{aligned}$$

so that

$$\begin{aligned} e_{b_{\Sigma v},k}^- &= e_{b_{\Sigma v},k-1}^+ \\ e_{b_{\Sigma\theta},k}^- &= e_{b_{\Sigma\theta},k-1}^+ \\ e_{b_{vel},k}^- &= e_{b_{vel},k-1}^+ \\ e_{b_{ac},k}^- &= e_{b_{ac},k-1}^+ \\ e_{b_{alt},k}^- &= e_{b_{alt},k-1}^+. \end{aligned}$$

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 61 of 73

3.4.2 State Estimation Error Covariance

Defining the estimation error covariance by P_k^- at time t_k prior to the state update and P_{k-1}^+ at time t_{k-1} after the state update and using the error equations in Eq. (45) and Eq. (46) we obtain the state error covariance propagation in Eq. (11) where the state estimation errors are

$$e_{k-1}^+ := \begin{bmatrix} e_{r,k-1} \\ e_{v,k-1} \\ e_{\theta,k-1} \\ e_{b_{\Sigma v,k-1}} \\ e_{b_{\Sigma \theta,k-1}} \\ e_{b_{\Delta v,k-1}} \\ e_{b_{\Delta \theta,k-1}} \\ e_{b_{\Delta t,k-1}} \end{bmatrix}^+ \quad \text{and} \quad e_k^- := \begin{bmatrix} e_{r,k} \\ e_{v,k} \\ e_{\theta,k} \\ e_{b_{\Sigma v,k}} \\ e_{b_{\Sigma \theta,k}} \\ e_{b_{\Delta v,k}} \\ e_{b_{\Delta \theta,k}} \\ e_{b_{\Delta t,k}} \end{bmatrix}^- \in \mathbb{R}^{22}$$

and the state transition matrix $\Phi_k \in \mathbb{R}^{22 \times 22}$ and process noise matrix $Q_k \in \mathbb{R}^{22 \times 22}$ are given by

$$\Phi_k = \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} \Delta t_k & -\frac{1}{2} \left(\hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] + \frac{1}{3} \hat{T}_{b,k-1}^i [(\Sigma \hat{\theta}_k^b \times \Sigma \hat{v}_k^b \times)] \right) \Delta t_k & -\hat{R}_a & \hat{R}_g & 0_{3 \times 3} & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & I_{3 \times 3} & -\left(\hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] + \frac{1}{2} \hat{T}_{b,k-1}^i [(\Sigma \hat{\theta}_k^b \times \Sigma \hat{v}_k^b \times)] \right) & -\hat{V}_a & \hat{V}_g & 0_{3 \times 3} & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & T(\Sigma \hat{\theta}_k) & 0_{3 \times 3} & -T_c^b & 0_{3 \times 3} & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 1 \end{bmatrix},$$

and


$$Q_k = M_k W M_k^T,$$

respectively, where

$$M_k = \begin{bmatrix} -\hat{R}_a & \hat{R}_g \\ -\hat{V}_a & \hat{V}_g \\ 0_{3 \times 3} & -T_c^b \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{22 \times 6}, \quad W = \begin{bmatrix} W_{\Delta v} & 0_{3 \times 3} \\ 0_{3 \times 3} & W_{\Delta \theta} \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

and

$$\begin{aligned} \hat{R}_a &= \frac{1}{2} \hat{T}_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{3} [\Sigma \hat{\theta}_k^b \times] \right) T_c^b \Delta t_k, & \hat{R}_g &= \frac{1}{6} \hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] T_c^b \Delta t_k, \\ \hat{V}_a &= \hat{T}_{b,k-1}^i \left(I_{3 \times 3} + \frac{1}{2} [\Sigma \hat{\theta}_k^b \times] \right) T_c^b, & \text{and} & \quad \hat{V}_g = \frac{1}{2} \hat{T}_{b,k-1}^i [\Sigma \hat{v}_k^b \times] T_c^b. \end{aligned}$$

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 62 of 73

4 Multi-mode Extended Kalman Filter

4.1 Mathematical model

The state vector is defined to be

$$\mathbf{x}_k := \begin{bmatrix} \mathbf{r}_{imu,k}^i \\ \mathbf{v}_{imu,k}^i \\ \mathbf{q}_i^c \\ \mathbf{b}_{\Sigma_{\Delta v}} \\ \mathbf{b}_{\Sigma_{\Delta \theta}} \\ \mathbf{b}_{vel} \\ \mathbf{b}_{ac} \\ b_{alt} \end{bmatrix} \in \mathbb{R}^{23}$$

The sensor biases, denoted by \mathbf{b}_{vel} , \mathbf{b}_{ac} , and b_{alt} , for the velocimeter, attitude camera, and altimeter, respectively, are modeled as random constants. One caveat to the state estimate update relates to the attitude update. Due to the fact that we implement a multiplicative update for the attitude quaternion, the state vector is \mathbb{R}^{23} , whereas the state estimation covariance is $\mathbb{R}^{22 \times 22}$. All of the preceding developments have focused on the attitude error in the form of small angles, denoted by $\delta\alpha$. This is done because the quaternion of rotation is subject to a unity norm constraint whereas the small angles are not. Therefore, if the portion of $\hat{\mathbf{x}}_k^+$ pertaining to the attitude is given by $\delta\hat{\alpha}_k^+$, then the quaternion update is

$$\hat{\mathbf{q}}_k^+ = \begin{bmatrix} \frac{1}{2}\delta\hat{\alpha}_k^+ \\ 1 \end{bmatrix} \otimes \hat{\mathbf{q}}_k^-, \quad (47)$$

where $\hat{\mathbf{q}}_k^-$ is the *a priori* estimate of the quaternion. Eq. (47) yields a quaternion which satisfies the unity norm constraint to first order, and so to ensure that the quaternion remains unity norm a brute-force normalization is performed such that $\hat{\mathbf{q}}_k^{+T} \hat{\mathbf{q}}_k^+ = 1$

4.2 mEKF Hierarchical Structure

A mEKF hierarchical structure (mEKFHS), shown in Figure 4 is proposed to represent a convenient vehicle for efficiently collecting data for input and output of the filter. The mEKFHS is comprised of five major substructures: meas, imuk, lload, Estimatekm, and Estimatekp. In addition, for simulation purposes, a Truth structure is proposed to allow a perfect navigation mode.

meask Structure The meask structure passes the measurement data to the mEKF. The structure is comprised of the measurement time tag, t_k , the data valid flag, and the measurements. The structure is detailed in Table 1.

imuk Structure The imuk structure passes the IMU Δv and $\Delta\theta$ data to the mEKF. Since the IMU internal rates are higher than the mEKF update rate, we allow for the IMU to bin multiple Δv and $\Delta\theta$ values and pass them as a group to the mEKF. The structure is comprised of a variable describing the number of Δv and $\Delta\theta$ values being passed and the actual Δv and $\Delta\theta$ values. The structure is detailed in Table 2.

Truth Structure The Truth structure is the connection to the environment and provides the pathway to the true states. In the actual implementation of the mEKF, the Truth structure would not exist. However, in the environment framework, it makes sense to have the capability to run a so-called perfect navigation mode. The Truth structure provides the necessary link to the environment to allow the mEKF to output a state estimate equivalent to the true state. The Truth structure is detailed in Table 3.


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 63 of 73

Table 1: The meask structure.


Sensor	Variable	Comment
Velocimeter	meask.vr_valid	1 = measurement valid; 0= measurement invalid
	meask.vr_time	Measurement time tag
	meask.vr	Velocimeter measurement (3-axes in sensor frame)
Attitude Camera	meask.ac_valid	1 = measurement valid; 0= measurement invalid
	meask.ac_time	Measurement time tag
	meask.ac	Attitude measurement (quaternion)
Attitude Camera	meask.alt_valid	1 = measurement valid; 0= measurement invalid
	meask.alt_time	Measurement time tag
	meask.alt	Altitude measurement (above reference sphere)
TRN	meask.trn_valid	1 = measurement valid; 0= measurement invalid
	meask.trn_time	Measurement time tag
	meask.trn	TRN measurement of position relative to ground feature

Table 2: The imuk structure.

Variable	Comment
imuk.nimu	Number of Δv and $\Delta \theta$ values being passed
imuk.deltav	Δv s in the IMU case frame
imuk.deltatheta	$\Delta \theta$ s in the IMU case frame

Table 3: The Truth structure.

Variable	Comment
Truth.t	Simulation time
Truth.r	The components of the vehicle position vector in the planet-centered inertial coordinate system
Truth.v	The components of the inertial vehicle velocity vector in the planet-centered inertial coordinate system
Truth.rcg	The vehicle center of gravity location along the body axes
Truth.a	Components of measurable (sensed) vehicle acceleration in the planet-centered inertial coordinate system
Truth.g	The gravity acceleration vector components along the planet-centered inertial coordinate system
Truth.ab	Truth IMU accelerations, components of measurable (sensed) acceleration in the body coordinate system w/ origin at the IMU
Truth.w	Truth IMU angular rates, vehicle angular velocity components in the body coordinate system w/ origin at the IMU.
Truth.vr	The vehicle velocity relative to the rotating planet
Truth.alt	The vertical altitude of the vehicle above the oblate planet
Truth.thr	The net thrust (vacuum thrust corrected for atmospheric backpressure effects)
Truth.lat	The geocentric latitude of the vehicle.
Truth.lng	The longitude of the vehicle measured east of the prime meridian
Truth.m	The current vehicle mass
Truth.q	Quaternion expressing the rotation from inertial to body

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 64 of 73

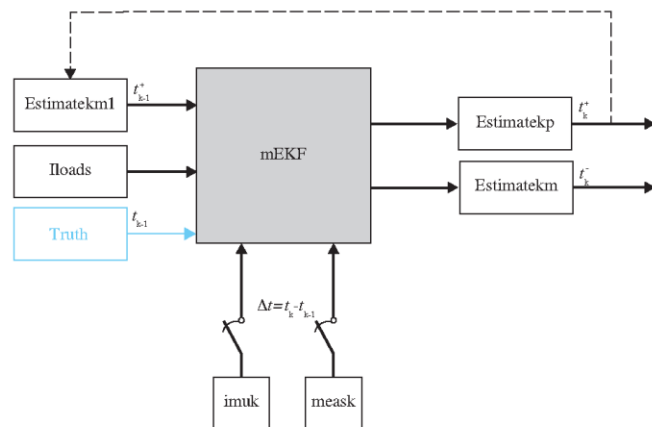


Figure 4: mEKF Hierarchical Structure

Iload Structure The Iload provides the required mEKF initialization data. Note that the initial state estimate and initial state error covariance matrix are initialized with the Estimatekm and Estimatekp structures. The Iload structure is detailed in Table 4.



	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 65 of 73

Table 4: The Iload structure.

Variable	Comment
Iload.nx	Number of states in the filter
Iload.np	Number of states in the covariance
Iload.nm	Number of measurements
Iload.nav_rate	Update rate of the mEKF in Hz
Iload.nav_rdt	Inverse of the navigation rate in sec
Iload.perfectnav	1 = Perfect nav is ON; 0 = mEKF is ON
Iload.gravitymodel	Only choice at current time is 'Central'
Iload.planet	Current choices are 'moon' or 'mars'
Iload.q_f_s	Orientation of planet-fixed reference frame with respect to the planet surface reference frame
Iload.r_ref_f	Location of the surface reference frame origin in the planet fixed-frame
Iload.omega_fi_f	Spin axis of moon at epoch
Iload.q_ie_f	Orientation of the moon inertial reference frame with respect to the planet-fixed reference frame at epoch
Iload.te	Epoch associated with q_ie_f in seconds
Iload.qimu	IMU process noise matrix
Iload.q tune	Tuning process noise matrix

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 66 of 73

5 Sensor Modeling

The mEKF navigation algorithm is a dual-state extended Kalman filter (EKF). The EKF is a model-based algorithm requiring models of the sensors. In general, there are two classes of sensor models required for an analysis of the six degree-of-freedom (6 DOF) integrated guidance, navigation, and control (GN&C) system. These are (1) high-fidelity models to support 6 DOF simulation of the sensors, and (2) sensor models for the navigation algorithm. This document is concerned only with sensor models for the navigation algorithm. To that end, a navigation sensor model includes the following:

1. a mathematical model represented by a nonlinear equation(s) as a function of the states of the system (such as position, velocity, attitude, etc.).
2. a measurement mapping matrix comprised of the partial derivatives (of the model above) evaluated at the most recent state estimate.
3. an error model comprised of random noise and systematic errors, including representative values of the uncertainty in the various error sources.

The sensors currently under consideration are: inertial measurement unit (IMU), spherical altimeter, velocimeter, and star camera.

5.1 Sensor hierarchical structure

A sensor hierarchical structure (SHS), shown in Figure 2 is proposed to represent a convenient vehicle for representing the multitude of sensors required for EDL. The SHS is comprised of four major substructures: Universe, Nucleus, Identity, and Errors.

Universe Structure The Universe structure provides the parameters required by the filter that describe the external environment. At the present time, the Universe structure models Mars and the Moon. For Mars we provide the polar radius, equatorial radius, J_2 , gravitational constant, rotation rate, and the inverse of the principal axes. The variables are described in Table 5. As the models increase in complexity, in addition to the parameters currently included, the Universe structure might also include, for example, any topographic maps, location and orientation of planetary bodies (such as the sun), and transformations from planet fixed to inertial reference frames.

Nucleus Structure The Nucleus structure describes the key elements of the sensor around which the model is constructed including, for example, the rate (internal), output rate, on/off events, and special processing (hardware specific) parameters. Currently, the Nucleus structure models the velocimeter, spherical altimeter, attitude camera, and IMU. Nominal values for each parameter is shown in Table 6.

Identity Structure The Identity structure describes the physical aspects of the sensor such as, for example, the spacecraft sensor layout (sensor locations, sensor reference frame orientations, etc.), and beam locations (for the velocimeter and altimeter).

Errors Structure The Errors structure includes the random and systematic errors associated with each sensor, such as biases, noise strengths, misalignments, and time constants.


	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 67 of 73

Table 5: Universe structure.

Name	Comment
Universe.mars.Ae	Radius of Mars for gravity calculations
Universe.mars.Apo	Polar radius of Mars
Universe.mars.Aeq	Equatorial radius of Mars
Universe.mars.J2	Gravity J2 term of Mars
Universe.mars.Mu	Gravitational constant of Mars
Universe.mars.Omega	Rotation rate of Mars
Universe.mars.prcpllengths	Computed value
Universe.moon.Ae	Radius of the Moon for gravity calculations
Universe.moon.Apo	Polar radius of the Moon
Universe.moon.Aeq	Equatorial radius of the Moon
Universe.moon.J2	Gravity J2 term of the Moon
Universe.moon.Mu	Gravitational constant of the Moon
Universe.moon.Omega	Rotation rate of the Moon
Universe.moon.prcpllengths	Computed value

Table 6: Nucleus structure.

Name	Comment
Sensor.Nucleus.velocimeter.edit_onoff	Velocimeter residual edit enabled/disabled
Sensor.Nucleus.velocimeter.edit_gate	Specification of probability gate (e.g., 9.14 → 99.75%)
Sensor.Nucleus.velocimeter.underweight_onoff	Velocimeter underweighting enabled/disabled (1/0)
Sensor.Nucleus.velocimeter.underweight_accept	Underweighting scale factor $0 < p \leq 1$
Sensor.Nucleus.velocimeter.rate	Velocimeter measurement rate in Hz
Sensor.Nucleus.altimeter.spherical.edit_onoff	Altimeter residual edit enabled/disabled (1/0)
Sensor.Nucleus.altimeter.spherical.edit_gate	Specification of probability gate
Sensor.Nucleus.altimeter.spherical.underweight_onoff	Altimeter underweighting enabled/disabled (1/0)
Sensor.Nucleus.altimeter.spherical.underweight_accept	Underweighting scale factor $0 < p \leq 1$
Sensor.Nucleus.altimeter.spherical.rate	Altimeter measurement rate in Hz
Sensor.Nucleus.attitudecamera.edit_onoff	Star camera residual edit enabled/disabled (1/0)
Sensor.Nucleus.attitudecamera.edit_gate	Specification of probability gate
Sensor.Nucleus.attitudecamera.underweight_onoff	Star camera underweighting enabled/disabled (1/0)
Sensor.Nucleus.attitudecamera.underweight_accept	Underweighting scale factor $0 < p \leq 1$
Sensor.Nucleus.attitudecamera.rate	Attitude measurement rate in Hz
Sensor.Nucleus.imu.deltav.thresh_limit	Tolerance for ΔV thresholding
Sensor.Nucleus.imu.deltav.thresh_onoff	IMU ΔV thresholding enabled/disabled (1/0)
Sensor.Nucleus.imu.deltav.thresh_sigsq	Standard deviation of expected IMU ΔV noise
Sensor.Nucleus.imu.deltatheta.thresh_limit	Tolerance for $\Delta \theta$ thresholding
Sensor.Nucleus.imu.deltatheta.thresh_onoff	IMU $\Delta \theta$ thresholding enabled/disabled (1/0)
Sensor.Nucleus.imu.deltatheta.thresh_sigsq	Standard deviation of expected IMU $\Delta \theta$ noise
Sensor.Nucleus.imu.rate	IMU internal rate in Hz
Sensor.Nucleus.imu.nimu	Number of buffered IMU data



	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 68 of 73

Table 7: Identity Structure.

Name	Comment
Sensor.Identity.altimeter.spherical.r.altimu_b	Location of the altimeter wrt the IMU represented in the body reference frame
Sensor.Identity.velocimeter.r.velimu_b	Location of the velocimeter wrt the IMU represented in the body reference frame
Sensor.Identity.attitudecamera.r.acimu_b	Location of the attitude camera wrt the IMU represented in the body reference frame
Sensor.Identity.q.c_b	Attitude quaternion of IMU case frame to the body reference frame
Sensor.Identity.q.a_b	Attitude quaternion of altimeter reference frame to the body reference frame
Sensor.Identity.q.ac_b	Attitude quaternion of attitude camera reference frame to the body reference frame
Sensor.Identity.q.vr_b	Attitude quaternion of velocimeter reference frame to the body reference frame

Table 8: Errors Structure.

Name	Comment
Sensor.Errors.velocimeter.bias	Velocimeter measurement bias
Sensor.Errors.velocimeter.rvel	Velocimeter measurement noise covariance matrix
Sensor.Errors.altimeter.spherical.bias	Spherical altimeter measurement bias
Sensor.Errors.altimeter.spherical.ralt	Spherical altimeter measurement noise covariance
Sensor.Errors.attitudecamera.bias	Attitude camera measurement bias
Sensor.Errors.attitudecamera.rac	Attitude camera measurement noise covariance matrix
Sensor.Errors.imu.accel_spec.bias	IMU Δv bias standard deviation ($1-\sigma$)
Sensor.Errors.imu.accel_spec.noise	IMU Δv noise covariance
Sensor.Errors.imu.gyro_spec.bias	IMU $\Delta \theta$ bias standard deviation ($1-\sigma$)
Sensor.Errors.imu.gyro_spec.noise	IMU $\Delta \theta$ noise covariance

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 69 of 73

For all four sensors, the data is nominally provided every Δt seconds computed via

$$\begin{aligned}
\text{Sensor.Nucleus.velocimeter.spec_dt} &= 1.0/\text{Sensor.Nucleus.velocimeter.rate} \\
\text{Sensor.Nucleus.altimeter.spherical.spec_dt} &= 1.0/\text{Sensor.Nucleus.altimeter.spherical.rate} \\
\text{Sensor.Nucleus.attitudecamera.spec_dt} &= 1.0/\text{Sensor.Nucleus.attitudecamera.rate} \\
\text{Sensor.Nucleus.imu.spec_dt} &= 1.0/\text{Sensor.Nucleus.imu.rate}
\end{aligned}$$

Also, we have

$$\begin{aligned}
\text{Universe.mars.prcplengths} &= \begin{bmatrix} \text{Universe.mars.Aeq} & 0 & 0 \\ 0 & \text{Universe.mars.Aeq} & 0 \\ 0 & 0 & \text{Universe.mars.Apo} \end{bmatrix}^{-1} \\
\text{Universe.moon.prcplengths} &= \begin{bmatrix} \text{Universe.moon.Aeq} & 0 & 0 \\ 0 & \text{Universe.moon.Aeq} & 0 \\ 0 & 0 & \text{Universe.moon.Apo} \end{bmatrix}^{-1}
\end{aligned}$$

5.2 Inertial Measurement Unit

With the sensor data structures defined as in the previous sections, we can represent the IMU with the Universe, Nucleus, Identity, and Error data structures. Recall that the model of the IMU was presented in Section 3.2.

5.2.1 IMU Sensor Hierarchical Structure

The IMU sensor hierarchical structure is shown in Figure 5. In Figure 5, we show the planet described in the Universe structure as Mars. Other planets can be represented. For example, the variable Universe.mars.Ae would be Universe.moon.Ae for the Moon.

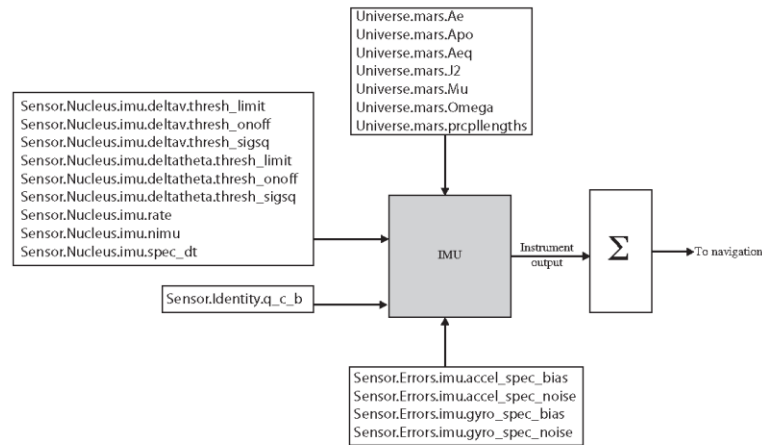



Figure 5: IMU Sensor Hierarchical Structure

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP- 09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 70 of 73

5.3 Altimeter

5.3.1 Mathematical model

The estimated spherical altimeter measurement at time t_k is

$$\hat{h}_k = \left[\hat{r}_{alt,k}^{i-} - r_{sph} \right] + \hat{b}_{alt,k}^{-},$$

where

$$\hat{r}_{alt,k}^{i-} = \left| \hat{\mathbf{r}}_{imu,k}^{i-} + \hat{\mathbf{T}}_{b,k}^{i-} \mathbf{r}_{alt/imu}^b \right|,$$

and

$$\hat{\mathbf{T}}_{b,k}^{i-} = \mathbf{T}^T(\hat{\mathbf{q}}_{t,k}^{b-}).$$

The value of r_{sph} is currently input as a constant in the spherical altimeter model, but it could be varied to capture the local topography, if desired and if a topography model were available to navigation. If an altitude measurement is given at time t_k by h_k , and the estimate is given by \hat{h}_k , then the spherical altimeter residual is

$$r_k = h_k - \hat{h}_k.$$

The measurement sensitivity for the spherical altimeter has the form

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0}_{1 \times 3} & \mathbf{H}_3 & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{H}_8 \end{bmatrix}.$$

The specific elements of the measurement sensitivity are

$$\begin{aligned} \mathbf{H}_1 &= \frac{1}{\left| \hat{\mathbf{r}}_{alt,k}^{i-} \right|} \hat{\mathbf{r}}_{alt,k}^{i-,T} \\ \mathbf{H}_3 &= -\frac{1}{\left| \hat{\mathbf{r}}_{alt,k}^{i-} \right|} \hat{\mathbf{r}}_{alt,k}^{i-,T} \hat{\mathbf{T}}_{b,k}^{i-} \mathbf{r}_{alt/imu}^b \\ \mathbf{H}_8 &= 1 \end{aligned}$$

where the estimated position of the altimeter in the inertial frame is

$$\hat{\mathbf{r}}_{alt,k}^{i-} = \hat{\mathbf{r}}_{imu,k}^{i-} + \hat{\mathbf{T}}_{b,k}^{i-} \mathbf{r}_{alt/imu}^b.$$

5.3.2 Altimeter Sensor Hierarchical Structure

The altimeter sensor hierarchical structure is shown in Figure 6.

5.4 Velocimeter


5.4.1 Mathematical model

The estimated relative velocity measurement is

$$\hat{\mathbf{v}}_{rel}^v = \mathbf{T}_b^v \hat{\mathbf{T}}_{i,k}^{b-} \left[\hat{\mathbf{v}}_{vel,k}^{i-} - \boldsymbol{\omega}_L^i \times \hat{\mathbf{r}}_{vel,k}^{i-} \right] + \hat{\mathbf{b}}_{vel,k}^{-},$$

where $\boldsymbol{\omega}_L^i$ is the rotation vector of the planet and the estimated position and velocity of the velocimeter in the inertial frame are, respectively,

$$\begin{aligned} \hat{\mathbf{r}}_{vel,k}^{i-} &= \hat{\mathbf{r}}_{imu,k}^{i-} + \hat{\mathbf{T}}_{b,k}^{i-} \mathbf{r}_{vel/imu}^b \\ \hat{\mathbf{v}}_{vel,k}^{i-} &= \hat{\mathbf{v}}_{imu,k}^{i-} + \hat{\mathbf{T}}_{b,k}^{i-} [\boldsymbol{\omega}_{b/i,k}^{b-} \times \mathbf{r}_{vel/imu}^b]. \end{aligned}$$

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 71 of 73

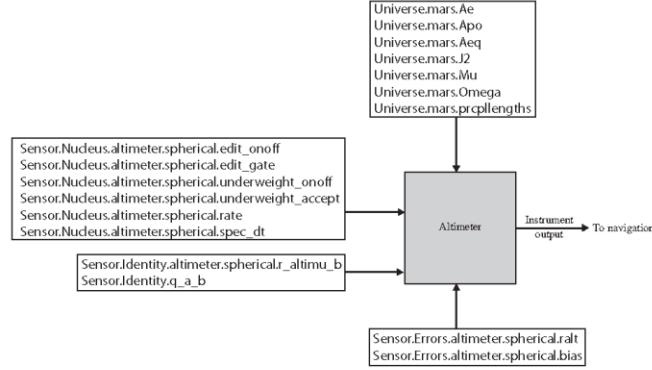


Figure 6: Altimeter Sensor Hierarchical Structure

With the relative velocity measurement given at time t_k by $\mathbf{v}_{rel,k}^v$, and the estimate given by $\hat{\mathbf{v}}_{rel,k}^v$, then the velocimeter residual is

$$\mathbf{r}_k = \mathbf{v}_{rel,k}^v - \hat{\mathbf{v}}_{rel,k}^v.$$

The measurement sensitivity for the relative velocimeter has the form

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{H}_6 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \end{bmatrix}.$$

The specific elements of the measurement sensitivity are

$$\begin{aligned} \mathbf{H}_1 &= -\mathbf{T}_b^v \hat{\mathbf{T}}_{i,k}^{b-} [\boldsymbol{\omega}_L^i \times] \\ \mathbf{H}_2 &= \mathbf{T}_b^v \hat{\mathbf{T}}_{i,k}^{b-} \\ \mathbf{H}_3 &= -\mathbf{T}_b^v \left[[\dot{\boldsymbol{\omega}}_{b/i,k}^{b-} \times \mathbf{r}_{vel/imu}^b] \times \right] + \mathbf{T}_b^v \hat{\mathbf{T}}_{i,k}^{b-} [\boldsymbol{\omega}_L^i \times] \hat{\mathbf{T}}_{b,k}^{i-} [\mathbf{r}_{vel/imu}^b \times] \\ &\quad + \mathbf{T}_b^v \left[\hat{\mathbf{T}}_{i,k}^{b-} [\hat{\mathbf{v}}_{vel,k}^{i-} - [\boldsymbol{\omega}_L^i \times \hat{\mathbf{r}}_{vel,k}^i]] \times \right] \\ \mathbf{H}_6 &= \mathbf{I}_{3 \times 3} \end{aligned}$$

5.4.2 Velocimeter Sensor Hierarchical Structure


The velocimeter sensor hierarchical structure is shown in Figure 7.

5.5 Attitude Camera

5.5.1 Mathematical model

The estimated quaternion attitude camera measurement is

$$\hat{\mathbf{q}}_{sr}^{ac} = \hat{\mathbf{q}}_{b,\eta,k}^{b-} \otimes \hat{\mathbf{q}}_b^{ac} \otimes \hat{\mathbf{q}}_{i,k}^{b-} \otimes \hat{\mathbf{q}}_{sr}^i,$$

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 72 of 73

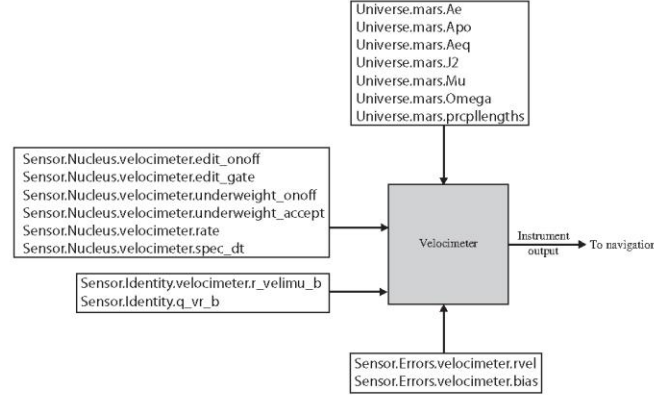


Figure 7: Velocimeter Sensor Hierarchical Structure

where

$$\hat{\bar{q}}_{b,\eta,k} = \begin{bmatrix} \sin\left(\frac{\hat{\bar{b}}_{ac,k}}{2}\right) \frac{\hat{\bar{b}}_{ac,k}}{\hat{\bar{b}}_{ac,k}} \\ \cos\left(\frac{\hat{\bar{b}}_{ac,k}}{2}\right) \end{bmatrix},$$

and $\hat{\bar{b}}_{ac,k} = \|\hat{\bar{b}}_{ac,k}\|$, where $\hat{\bar{b}}_{ac,k}$ is the estimate of the attitude camera bias described in Section 3.4.1. Therefore, if a measurement of the quaternion attitude camera is given at time t_k by $\hat{q}_{sr,k}^{ac}$, and the estimate is given by $\hat{q}_{sr,k}^{ac}$, then the attitude camera residual is

$$\mathbf{r}_k = 2 \left(\hat{q}_{sr,k}^{ac} \mathbf{q}_{sr,k}^{ac} + \mathbf{q}_{sr,k}^{ac} \times \hat{q}_{sr,k}^{ac} - q_{sr,k}^{ac} \hat{q}_{sr,k}^{ac} \right).$$

The measurement sensitivity for a quaternion attitude camera measurement has the form

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{H}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{H}_7 & \mathbf{0}_{3 \times 1} \end{bmatrix}.$$

The specific elements of the measurement sensitivity are


$$\begin{aligned} \mathbf{H}_3 &= \hat{\mathbf{T}}_{b,\eta,k}^{-} \mathbf{T}_b^{ac} \\ \mathbf{H}_7 &= \mathbf{I}_{3 \times 3}, \end{aligned}$$

where $\hat{\mathbf{T}}_{b,\eta,k}^{-}$ is the transformation matrix created by the estimated bias-noise quaternion,

$$\hat{\mathbf{T}}_{b,\eta,k}^{-} = \mathbf{T}(\hat{\bar{q}}_{b,\eta,k}).$$

5.5.2 Attitude Camera Sensor Hierarchical Structure

The attitude camera sensor hierarchical structure is shown in Figure 8.

	NASA Engineering and Safety Center Technical Assessment Report	Document #: NESC-RP-09-00530	Version: 1.0
Title: Phase 2: Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis			Page #: 73 of 73

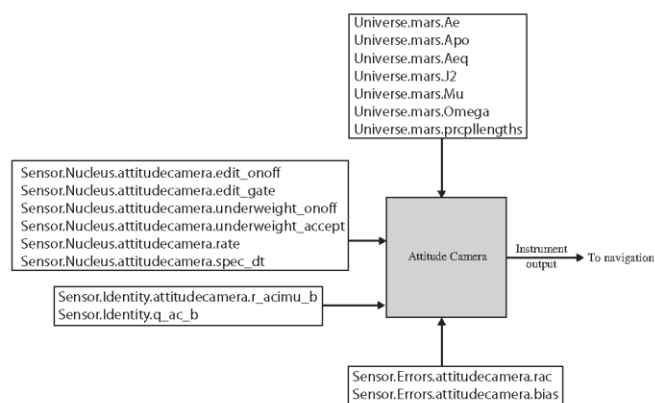


Figure 8: Attitude Camera Sensor Hierarchical Structure

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>						
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)		
01-02 - 2011		Technical Memorandum		December 2009 - February 2011		
4. TITLE AND SUBTITLE Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis, Phase 2 Results				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Murri, Daniel G.				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER 869021.05.07.01.07		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199				8. PERFORMING ORGANIZATION REPORT NUMBER L-19995 NESC-RP-09-00530		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001				10. SPONSOR/MONITOR'S ACRONYM(S) NASA		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/TM-2011-217063		
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 16-Space Transportation and Safety Availability: NASA CASI (443) 757-5802						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT The NASA Engineering and Safety Center (NESC) was requested to establish the Simulation Framework for Rapid Entry, Descent, and Landing (EDL) Analysis assessment, which involved development of an enhanced simulation architecture using the Program to Optimize Simulated Trajectories II simulation tool. The assessment was requested to enhance the capability of the Agency to provide rapid evaluation of EDL characteristics in systems analysis studies, preliminary design, mission development and execution, and time-critical assessments. Many of the new simulation framework capabilities were developed to support the Agency EDL-Systems Analysis (SA) team that is conducting studies of the technologies and architectures that are required to enable human and higher mass robotic missions to Mars. The findings, observations, and recommendations from the NESC are provided in this report.						
15. SUBJECT TERMS NASA Engineering and Safety Center; Entry, Descent, and Landing; Program to Optimize Simulated Trajectories II; Rapid EDL Analysis Simulation						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)	
U	U	U	UU	78	19b. TELEPHONE NUMBER (Include area code) (443) 757-5802	